BAYESIAN METHODS AND CONFIDENCE INTERVALS FOR AUTOMATIC TARGET RECOGNITION OF SAR CANONICAL SHAPES

THESIS

RICHARD W. RADEMACHER, Captain, USAF

AFIT-ENG-14-M-62

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

AFIT-ENG-14-M-62

BAYESIAN METHODS AND CONFIDENCE INTERVALS
FOR AUTOMATIC TARGET RECOGNITION OF SAR CANONICAL SHAPES

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Electrical Engineering

RICHARD W. RADEMACHER, B.S.E.E.
Captain, USAF

March 2014

AFIT-ENG-14-M-62

BAYESIAN METHODS AND CONFIDENCE INTERVALS FOR AUTOMATIC TARGET RECOGNITION OF SAR CANONICAL SHAPES

RICHARD W. RADEMACHER, B.S.E.E.
Captain, USAF

Approved:

//signed// 11 Mar 2014
Dr. Julie A. Jackson, PhD (Chairman) Date

//signed// 14 Mar 2014
Capt. Dustin G. Mixon, PhD, USAF (Member) Date

//signed// 12 Mar 2014
Dr. Christine M. Schubert Kabban, PhD (Member) Date

AFIT-ENG-14-M-62

Abstract

This research develops a new Bayesian technique for the detection of scattering primitives in synthetic aperture radar (SAR) phase history data received from a sensor platform. The primary goal of this research is the estimation of size, position, and orientation parameters defined by the "canonical" shape primitives of Jackson. Previous Bayesian methods for this problem have focused on the traditional *maximum a posteriori* (MAP) estimate based on the posterior density. A new concept, the probability mass interval, is developed. In this technique the posterior density is partitioned into intervals, which are then integrated to form a probability mass over that interval using the Gaussian quadrature numerical integration techniques. The posterior density is therefore discretized in such a way that the location of local peaks are preserved. A formal treatment is given to the effect of locally integrating the posterior density in the context of parameter estimation. It is shown that the operation of choosing the interval with the highest probability mass is equivalent to an optimum Bayesian estimator that places zero cost on a "range" of parameters. The range is user-controlled, and is akin to the idea of parameter resolution. Additionally the peak-preserving property allows the user to begin with coarse intervals and "zoom" in as they see fit. Associated with these estimates is a measure of quality called the credible interval (or credible set). The credible interval (set) is a region of parameter space where the "true" parameter is located with a user-defined probability. Narrow credible intervals are associated with high-quality estimates while wide credible intervals are associated with poor estimates. The techniques are implemented in state-of-the-art graphics processor unit (GPU) hardware, which allows the numerical integration to be performed in a reasonable time. A typical estimator requires several hundred million computations and the GPU implementation reduces the computation time from several hours to a few seconds. The mass interval estimation technique may be used on any

Bayesian problem, but is demonstrated here using each of the canonical shape models of Jackson. The technique successfully estimates parameters in different scenarios including simple shapes, multiple shapes, incorrect shape (i.e. trying to estimate parameters using the wrong model). The results of this research are a new exploration of the posterior distributions of the canonical shape model, improved numerical integration strategies, and a new statistical technique for the Bayesian estimation of parameters.

This thesis is dedicated to to all the teachers I've ever had; to my mother and father, who were the first; and to failure, which is the best.

Acknowledgments

Thanks to all the members of the Radar Instrumentation Lab (RAIL) for their help and support.

A very special thanks to my advisor Dr. Julie Jackson and committee members Capt. Mixon and Dr. Schubert Kabban, for the freedom to pursue blind alleys, re-derive the obvious, for the guidance when I was in over my head, and for the frequent realignments necessary to ensure this research was ultimately completed.

This thesis research was sponsored by the Air Force Office of Scientific Research under lab task LRIR12RY19COR.

RICHARD W. RADEMACHER

Table of Contents

Page

Page

List of Figures

Figure Page

List of Tables

List of Symbols

Symbol	Definition
$\underline{\boldsymbol{v}}$	A vector with elements v_i
$\underline{\underline{\boldsymbol{A}}}$	A matrix with elements a_{ij}
$\boldsymbol{E}[\bullet]$	Expected Value
$\boldsymbol{O}[P]_i$	i^{th} largest member of an ordered permutation of a set P
$\boldsymbol{p}(\bullet)$	Probability Density Function
$\boldsymbol{P}(\bullet)$	Probability Mass Function
$\boldsymbol{p}(A, B)$	Probability (density or mass) of both A and B occurring
$\boldsymbol{p}(A \mid B)$	Conditional probability (density or mass) of event A occurring given event B
$\mathbb{R}$	The set of all real numbers
$\mathbb{N}$	The set of natural numbers
j	The imaginary number $j = \sqrt{-1}$
$\underline{\boldsymbol{v}} \bullet \underline{\boldsymbol{w}}$	The dot product of vectors $\underline{\boldsymbol{v}}$ and $\underline{\boldsymbol{w}}$
$\langle a, b\rangle_w$	The inner product of polynomials $a(x)$ and $b(x)$ under weight $w(x)$
$\underline{\boldsymbol{k}}$	The vector form of the wavenumber
$\underline{\boldsymbol{r}}$	The target range vector
$\mathfrak{Re}(x)$	Real part of a scalar (or vector) x
$\mathfrak{Im}(x)$	Imaginary part of a scalar (or vector) x
$f^{(n)}(x)$	n-th derivative of $f(x)$
$\lvert f(x)\rvert$	Absolute value of the function $f(x)$
δ_{ij}	Kronecker delta: $\delta_{ij} = 1$ if $i = j$, 0 if $i \neq j$
$\operatorname{sinc} x$	The value $\sin x/x$ where $\operatorname{sinc} 0 = 1$
$\mathcal{C}$	A closed region in some abstract space
$\emptyset$	The empty set

Symbol	Definition
$x_i \sim \boldsymbol{p}(X)$	A sample x_i drawn from a distribution $\boldsymbol{p}(X)$

List of Acronyms

Acronym	Definition
AFIT	Air Force Institute of Technology
ATR	Automatic Target Recognition
AWGN	Additive White Gaussian Noise
CPU	Central Processing Unit
GPU	Graphics Processing Unit
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
IID	Independent, Identically Distributed
ISAR	Inverse Synthetic Aperture Radar
MLE	Maximum Likelihood Estimate
MAP	Maximum *a posteriori*
PDF	Probability Density Function
PTX	CUDA Portable Executable File
RCS	Radar Cross Section
SAR	Synthetic Aperture Radar

BAYESIAN METHODS AND CONFIDENCE INTERVALS
FOR AUTOMATIC TARGET RECOGNITION OF SAR CANONICAL SHAPES

I. Introduction

Radar systems provide a means for high-fidelity sensing of a target scene from a remote location. Depending on the choice of carrier frequency, a radar system can provide sensing in environmental conditions that would inhibit optical sensing, such as poor weather, nighttime observation, and (for very low carrier frequencies) beyond line-of-site observations.

A single radar observation is inherently 1-dimensional, and range to target is the simplest measurement. Synthetic aperture radar (SAR) is a more sophisticated technique whereby the sensing platform (be it an aircraft, ship or vehicle) takes multiple individual radar measurements of a stationary target as it moves along a path. If the path has sufficient diversity, meaning each new observation adds more target information, the collection of observations can be computer-processed to provide a 2-dimensional (or 3-dimensional) measurement of the target scene. Alternatively, when the sensor is stationary and the target moves, the process is known as inverse SAR or ISAR. When the measurements are processed to estimate target reflectivity, the process is known as *SAR imaging*.

SAR imaging has proven to be an extremely successful tool. Modern SAR systems occupy large bandwidths with high dynamic range. As a result, very high-resolution and detailed images are available. This is a great improvement over early radars, but places large demands on human analysts to detect interesting targets from the images. Often the SAR images bear little resemblance to their optical counterparts and the analyst must be very skilled to recognize interesting targets amongst other uninteresting ones.

To this end, an open area of research is the problem of automatic target recognition (ATR) whereby a computer recognizes known targets from the radar data. The problem of accurately recognizing complex shapes is a very active research topic since the information gained by a SAR platform is flight-path dependent and incomplete. The processing of raw SAR data (the so-called *phase history*) into SAR images may provide data that is most intuitive to the human analyst but forming such images is a computationally intensive process and may be unnecessary for computer-assisted ATR detection.

One of the first problems to be solved for a successful ATR system is identifying and recognizing characteristic features of the target. Jackson [1] developed a method of predicting the received phase history of a series of standard shape types: plates, dihedrals (right-angle bracket), spheres, trihedrals (corner brackets), top-hats (a right angle revolved about an axis), and cylinders. These so-called 'canonical' shapes are shown in Figure 2.1. Each shape has been parametrized by three classes of parameters: *size* parameters (length, width, height, radius, etc.), *pose* parameters (roll, pitch, yaw) and *position* parameters (X, Y, Z position). Size parameters vary by shape type.

1.1 Problem Description

Using the Jackson model, we are presented with the problem of extracting the shape type and its parameters from a received phase history. When presented with a SAR phase history, we wish to deliver to the ATR software a list of detected shapes, their types, locations, sizes and orientation. The user must have the freedom to decide at what fidelity they wish the detector to operate. The user must also be provided with some measure of the confidence we have in those estimates. We wish to do this in the presence of additive noise.

1.2 Research Goals, Methodology, and Results

The goal of this research is twofold. First we will attempt to use Bayesian methods to estimate parameters and shape types from an unknown phase history. Secondly, a calculation of the credibility of that estimate will be performed. The result is a dataset of grid points in parameter space that meets a specified confidence metric.

We approach these goals as follows. First, we develop a scheme to reduce the infinitely dense posterior distribution into a discrete set of probability mass points. We show this technique can be used at coarse, and then progressively finer, resolutions without loss of the important notion of peak location. Secondly, we adapt well-known numerical methods to the problem of integrating unknown functions (such as the posterior distribution function) over finite bounds. We will show that this requires high computation power and describe the adaptation of the Jackson [1] phase history model to low-cost modern GPU processors. Finally a series of test cases will be applied to the algorithm to gauge performance.

The results of these tests show that the multi-zoom technique is effective at providing fast, low precision estimates of the shape parameters, as well as high-precision estimates over a much smaller interval. The technique is also effective for multiple shapes (provided the number of shapes is known a priori). Discretizing the posterior density is also an effective visualization technique and allows the analyst to explore degenerate cases (such as the effects of using the wrong shape model in Section 4.4).

1.3 Thesis Organization

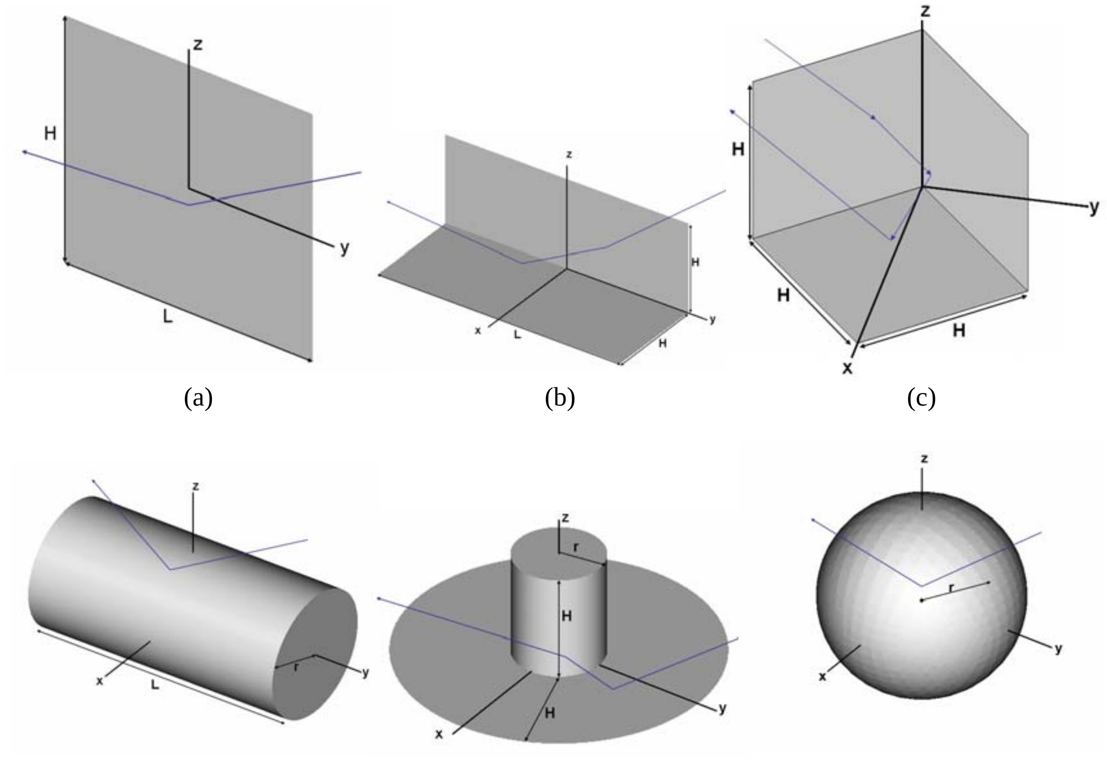
The thesis is organized as follows: Chapter 2 covers background material, including the data model, basic definitions and commonly used methods and prior work on the topic. Chapter 3 covers the new methods and techniques that were developed over the course of this research. Chapter 4 describes the results of applying these new methods to a series of representative simulations of real-world data, and also highlights the observations uncovered that gain insight into the strengths and weaknesses of the techniques. Chapter 5 concludes the research and summarizes pertinent results as well as suggestions for future topics of research.

Note: throughout this thesis the graphs and figures are annotated with a small identifier starting with the phrase 'guid' followed by a number. This references a file that has the original MATLAB source code used to generate that graph, which is available in the AFIT archives for this thesis.

II. Background

2.1 Canonical Shape Models

The canonical shape model used in this research follows the work of [1, 2]. The model defines a bistatic, polarization-dependent prediction of the SAR phase history for six standard shapes: plate, dihedral, trihedral, sphere, cylinder and top-hat. These are shown in Figure 2.1.



(a) (b) (c)

(d) (e) (f)

Figure 2.1: Canonical shapes. (a) Plate (b) Dihedral (c) Trihedral (d) Cylinder (e) Top-hat (f) Sphere. Used with permission from [1].

The Jackson model for a single polarization factors the phase history into two parts: a complex shape-, size- and pose-dependent magnitude[1] factor M and a complex position-dependent phase factor. This model is bistatic, and so it is dependent on positions of the transmitter and receiver. Each shape has a series of different parameters which we will label generically as $\underline{\mathbf{\Theta}}$. The convention used here is to let the transmitter azimuth, elevation, and range be labeled $\{\theta_t, \phi_t, R_t\}$ and the corresponding receiver azimuth, elevation, and range be labeled $\{\theta_r, \phi_r, R_r\}$. The resulting phase history as a function of wavenumber $k = \frac{2\pi f}{c}$ is given by

$$S(k) = M(k, \theta_t, \phi_t, \theta_r, \phi_r, \underline{\mathbf{\Theta}}) e^{-jk\left(R_t + R_r + \Delta R(\underline{\mathbf{\Theta}})\right)}. \tag{2.1}$$

In this research we will utilize phase history vectors that are the result of sampling Equation (2.1) along a series of K azimuth and elevation points

$$\underline{\boldsymbol{S}}(k) = \begin{bmatrix} M(k, \theta_{t_1}, \phi_{t_1}, \theta_{r_1}, \phi_{r_1}, \underline{\mathbf{\Theta}}) e^{-jk\left(R_{t_1} + R_{r_1} + \Delta R_1(\underline{\mathbf{\Theta}})\right)} \\ M(k, \theta_{t_2}, \phi_{t_2}, \theta_{r_2}, \phi_{r_2}, \underline{\mathbf{\Theta}}) e^{-jk\left(R_{t_2} + R_{r_2} + \Delta R_2(\underline{\mathbf{\Theta}})\right)} \\ \vdots \\ M(k, \theta_{t_K}, \phi_{t_K}, \theta_{r_K}, \phi_{r_K}, \underline{\mathbf{\Theta}}) e^{-jk\left(R_{t_K} + R_{r_K} + \Delta R_K(\underline{\mathbf{\Theta}})\right)} \end{bmatrix}. \tag{2.2}$$

Several of the canonical shapes possess a phase center that does not coincide with its geometric centroid. This artificial phase center creates a parameter and shape dependent range offset $\Delta R(\underline{\mathbf{\Theta}})$. The phase centers for the cylinder, top-hat, and sphere are given in Equations (2.11), (2.13), and (2.15). For cases without an artificial phase center, the true range to the shape centroid (X, Y, Z) is given in [1] as

$$\begin{aligned} R_t + R_r \approx\ & X(\cos\phi_t \cos\theta_t + \cos\phi_r \cos\theta_r) \\ & + Y(\sin\phi_t \cos\theta_t + \sin\phi_r \cos\theta_r) \\ & + Z(\sin\theta_t + \sin\theta_r). \end{aligned} \tag{2.3}$$

[1] ‘Magnitude’ is used loosely to differentiate this factor from the position phase factor. The magnitude factor may be complex and non-positive.

2.1.1 Coordinate System.

The Jackson phase history models [1] utilize a geometric transformation to describe the orientation of a shape in terms of local angles θ_t (the elevation angle from transmitter to shape), θ_r (elevation angle from receiver to shape), ϕ_t (azimuth angle from transmitter to shape), and ϕ_r (azimuth angle from receiver to shape).

Throughout the simulations, we rely on a intuitive notion of the Euler angles roll, pitch, and yaw. These are related to the local angles by use of the transformations given in [1] (Sec 3.4.7) and in [3].

2.1.2 Plate.

The canonical plate shape is a phase history model for an infinitely thin rectangular reflector. The basic shape parameters are its length L, height H and Radar Cross Section (RCS) A along with the position X, Y, Z and roll-pitch-yaw pose angle. The complex magnitude M_{plate} is given in [1] as

$$\begin{aligned} M_{plate} = A\left(\frac{jk}{\sqrt{\pi}}\right) \operatorname{sinc}\left(k\frac{L}{2}(\sin\phi_t\cos\theta_t + \sin\phi_r\cos\theta_r)\right) \\ \times \operatorname{sinc}\left(k\frac{H}{2}(\sin\theta_t + \sin\theta_r)\right) \\ \theta_t, \theta_r, \phi_t, \phi_r \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]. \end{aligned} \tag{2.4}$$

The phase center of the canonical plate coincides with its geometric center. The corresponding range offset in Equation (2.1) is

$$\Delta R_{plate} = 0. \tag{2.5}$$

2.1.3 Dihedral.

The canonical dihedral shape is a phase history model for an infinitely thin right-angle reflector. The right angle is composed of two electrically connected flat plates of height H joined on the length L axis. The basic shape parameters $\underline{\boldsymbol{\Theta}}$ are its length L, height H and RCS A along with the position X, Y, Z and roll-pitch-yaw pose angle. The complex magnitude $M_{dihedral}$ is given in [1] as

$$\begin{aligned} M_{dihedral} &= A\left(\frac{jk}{\sqrt{\pi}}\right)\operatorname{sinc}\left(k\frac{L}{2}(\sin\phi_t\cos\theta_t + \sin\phi_r\cos\theta_r)\right) \\ &\times \operatorname{sinc}\left(kH(\cos\theta_t - \cos\theta_r)\right) \times \begin{cases} \cos\frac{\theta_t+\theta_r}{2} & \theta_t,\theta_r \in [0,\frac{\pi}{4}] \\ \sin\frac{\theta_t+\theta_r}{2} & \theta_t,\theta_r \in [\frac{\pi}{4},\frac{\pi}{2}] \end{cases} \\ &\qquad \phi_t,\phi_r \in \left[-\frac{\pi}{2},\frac{\pi}{2}\right] \end{aligned} \tag{2.6}$$

The phase center of the canonical dihedral coincides with its geometric center. The corresponding range offset in Equation (2.1) is

$$\Delta R_{dihedral} = 0. \tag{2.7}$$

2.1.4 Trihedral.

The canonical trihedral shape is a phase history model for an infinitely thin right-corner reflector. The corner is composed of three electrically connected flat plates of height H and width H (since these are the same, we refer to the shape as only having a 'height' parameter but not a 'width'). The basic shape parameters $\underline{\mathbf{\Theta}}$ are its length height H and RCS A along with the position X, Y, Z and roll-pitch-yaw pose angle. The complex magnitude $M_{trihedral}$ is given in [1] as

$$\begin{aligned} M_{trihedral} &= \frac{jk}{\sqrt{\pi}}A\operatorname{sinc}\Big[kH(\cos\theta_t - \cos\theta_r)\Big] \\ &\times \left\{ \begin{array}{ll} \sin\left(\frac{\theta_t+\theta_r}{2} + \frac{\pi}{4} - \tan^{-1}\frac{1}{\sqrt{2}}\right), & \theta_r \in \left[0, \tan^{-1}\frac{1}{\sqrt{2}}\right] \\ \cos\left(\frac{\theta_t+\theta_r}{2} + \frac{\pi}{4} - \tan^{-1}\frac{1}{\sqrt{2}}\right), & \theta_r \in \left[\tan^{-1}\frac{1}{\sqrt{2}}, \frac{\pi}{2}\right] \end{array} \right\} \\ &\times \frac{1}{2}\Bigg[\operatorname{sinc}\Big[kH\Big(\cos\Big(\phi_r - \frac{\pi}{4}\Big)\cos\theta_r - \cos\Big(\phi_t - \frac{\pi}{4}\Big)\cos\theta_t\Big)\Big] \\ &\qquad + \operatorname{sinc}\Big[kH\Big(\cos\Big(\phi_r + \frac{\pi}{4}\Big)\cos\theta_r - \cos\Big(\phi_t + \frac{\pi}{4}\Big)\cos\theta_t\Big)\Big]\Bigg] \\ &\times \left\{ \begin{array}{ll} -\cos\left(\frac{\phi_t+\phi_r}{2} - \frac{\pi}{4}\right), & \phi_r \in \left[-\frac{\pi}{4}, 0\right] \\ \sin\left(\frac{\phi_t+\phi_r}{2} - \frac{\pi}{4}\right), & \phi_r \in \left[0, \frac{\pi}{4}\right] \end{array} \right\}. \end{aligned} \tag{2.8}$$

The phase center of the canonical trihedral coincides with its geometric center. The corresponding range offset in Equation (2.1) is

$$\Delta R_{trihedral} = 0. \tag{2.9}$$

2.1.5 Cylinder.

The canonical cylinder shape is a phase history model for a cylindrical drum. The basic shape parameters $\underline{\boldsymbol{\Theta}}$ are its length L, radius R, and RCS A along with the position X, Y, Z and roll-pitch-yaw pose angle. The complex magnitude $M_{cylinder}$ is given in [1] as

$$\begin{aligned} M_{cylinder} &= \sqrt{\frac{jk}{\cos\phi_t}} A \cos\phi_r \operatorname{sinc}\left[k\frac{L}{2}(\sin\phi_t\cos\theta_t + \sin\phi_r\cos\theta_r)\right] \\ &\theta_t, \theta_r \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]. \end{aligned} \tag{2.10}$$

The phase center of the canonical cylinder does not coincide with its geometric center due to the curvature of the drum. The corresponding range offset in Equation (2.1) is

$$\Delta R_{cylinder} = R\cos\left(\frac{\theta_t - \theta_r}{2}\right)(\cos\phi_t + \cos\phi_r). \tag{2.11}$$

2.1.6 Top-hat.

The canonical top-hat shape is a phase history model for a drum of radius R with an infinitely thin large disk at the base. In the case of zero roll-pitch-yaw pose, this is equivalent to the circular revolution of a right angle about the z axis. The height of the drum is equal to the 'brim' of the top-hat, thus the revolved right angle is two plates of equal length. Equivalently, the disk is of radius $R + H$. The basic shape parameters $\underline{\boldsymbol{\Theta}}$ are its height H, radius R and RCS A along with the position X, Y, Z and roll-pitch-yaw pose angle. The complex magnitude M_{tophat} is given in [1] as

$$M_{tophat} = A\sqrt{jk}\,[kH(\cos\theta_t - \cos\theta_r)] \begin{cases} \sin\left(\frac{\theta_t+\theta_r}{2}\right), & \theta_t, \theta_r \in [0, \frac{\pi}{4}] \\ \cos\left(\frac{\theta_t+\theta_r}{2}\right), & \theta_t, \theta_r \in [\frac{\pi}{4}, \frac{\pi}{2}] \end{cases}. \tag{2.12}$$

The phase center of the canonical top-hat does not coincide with its geometric center due to the curvature of the drum, similar to that of the cylinder. The corresponding range offset in Equation (2.1) is

$$\Delta R_{tophat} = R\cos\left(\frac{\phi_t - \phi_r}{2}\right)(\cos\theta_t + \cos\theta_r). \tag{2.13}$$

2.1.7 Sphere.

The canonical sphere is a phase history model for a perfect ball shape. For a sphere located at $X = Y = Z = 0$ the sphere looks the same from all angles. The basic shape parameters $\underline{\mathbf{\Theta}}$ are its radius R and RCS A along with the position X, Y, Z and roll-pitch-yaw pose angle. The complex magnitude M_{sphere} is given in [1] as

$$M_{sphere} = A\sqrt{\pi}. \tag{2.14}$$

The phase center of the canonical sphere does not coincide with its geometric center due to its curvature in two planes. The corresponding range offset in Equation (2.1) is

$$\begin{aligned}\Delta R_{sphere} = R \Bigg[&(\cos\theta_t + \cos\theta_r)\cos\left(\frac{\phi_t - \phi_r}{2}\right)\cos\left(\frac{\theta_t + \theta_r}{2}\right) \\ &+ (\sin\theta_t + \sin\theta_r)\sin\left(\frac{\theta_t + \theta_r}{2}\right)\Bigg].\end{aligned} \tag{2.15}$$

2.2 Bayesian Parameter Estimation

The models in the previous section allow software to predict the phase history if the parameters (X, Y, Z, roll, pitch, yaw, length, height, radius) are known. The goal of this thesis is the opposite problem: to detect and estimate the parameters given only the phase history. The Jackson model provides the basis for this effort.

Consider an ATR system whereby some prior knowledge of likely canonical shapes is provided. This may be due to ‘favorite’ target shapes that the system is particularly interested in, or key ‘signature’ shapes that may complete a composite target, or some other reason. Incorporating this prior knowledge should improve detection results. In this section, we review some common results from statistical estimation from an abstract point of view. In Chapter 3, we will apply these results to the detection of canonical shapes.

Let the collection of all the parameters for a given shape be the D-dimensional vector denoted $\underline{\mathbf{\Theta}}$, and let a particular phase history (flattened into a vector) be denoted $\underline{\boldsymbol{y}}$. Consider the estimation of a parameter $\underline{\mathbf{\Theta}}$ given an observation $\underline{\boldsymbol{y}}$. From basic probability theory, the relations hold for two events A and B (from [4], Ch. 3)

$$\boldsymbol{p}(A,B) = \boldsymbol{p}(A\,|\,B)\boldsymbol{p}(B), \tag{2.16}$$

$$\boldsymbol{p}(A\,|\,B)\boldsymbol{p}(B) = \boldsymbol{p}(B\,|\,A)\boldsymbol{p}(A), \tag{2.17}$$

$$\boldsymbol{p}(A) = \int_{\mathbb{R}} \boldsymbol{p}(A,B)dB. \tag{2.18}$$

For an observation $\underline{\boldsymbol{y}}$ and parameter $\underline{\boldsymbol{\Theta}}$ the conditional probability $\boldsymbol{p}(\underline{\boldsymbol{y}}\,|\,\underline{\boldsymbol{\Theta}})$ is called the *likelihood* of $\underline{\boldsymbol{y}}$ and represents the probability of a particular observation if the parameters $\underline{\boldsymbol{\Theta}}$ were known exactly. A ‘best guess’ for $\underline{\boldsymbol{\Theta}}$ would be one that gives the highest probability of $\boldsymbol{p}(\underline{\boldsymbol{y}}\,|\,\underline{\boldsymbol{\Theta}})$, and is known as the *maximum likelihood estimate* (MLE) [5]

$$\underline{\boldsymbol{\Theta}}_{MLE} = \arg\max_{\underline{\boldsymbol{\Theta}}} \boldsymbol{p}(\underline{\boldsymbol{y}}\,|\,\underline{\boldsymbol{\Theta}}). \tag{2.19}$$

Now consider a situation where information on the probability of a particular parameter value $\underline{\boldsymbol{\Theta}}$ occurring (regardless of any observation) is known ahead of time. The *prior distribution* $\boldsymbol{p}(\underline{\boldsymbol{\Theta}})$ can be used with the likelihood by means of Equation (2.16) to yield the famous Bayes’ Rule

$$\boldsymbol{p}(\underline{\boldsymbol{\Theta}}\,|\,\underline{\boldsymbol{y}}) = \frac{\boldsymbol{p}(\underline{\boldsymbol{y}}\,|\,\underline{\boldsymbol{\Theta}})\boldsymbol{p}(\underline{\boldsymbol{\Theta}})}{\boldsymbol{p}(\underline{\boldsymbol{y}})}. \tag{2.20}$$

Often the likelihood function is known from the observed data, but the probability of observation $\boldsymbol{p}(\underline{\boldsymbol{y}})$ (the denominator of Equation (2.20)) is not. The denominator $\boldsymbol{p}(\underline{\boldsymbol{y}})$ is simply the *marginalized* joint density $\boldsymbol{p}(\underline{\boldsymbol{y}},\underline{\boldsymbol{\Theta}})$. By Equation (2.18)) [6] and Bayes’ Rule can be rewritten as

$$\begin{aligned}\boldsymbol{p}(\underline{\boldsymbol{\Theta}}\,|\,\underline{\boldsymbol{y}}) &= \frac{\boldsymbol{p}(\underline{\boldsymbol{y}}\,|\,\underline{\boldsymbol{\Theta}})\boldsymbol{p}(\underline{\boldsymbol{\Theta}})}{\int_{\mathbb{R}^D} \boldsymbol{p}(\underline{\boldsymbol{y}},\underline{\boldsymbol{\Theta}})d\underline{\boldsymbol{\Theta}}} \\ &= \frac{\boldsymbol{p}(\underline{\boldsymbol{y}}\,|\,\underline{\boldsymbol{\Theta}})\boldsymbol{p}(\underline{\boldsymbol{\Theta}})}{\int_{\mathbb{R}^D} \boldsymbol{p}(\underline{\boldsymbol{y}}\,|\,\underline{\boldsymbol{\Theta}})\boldsymbol{p}(\underline{\boldsymbol{\Theta}})d\underline{\boldsymbol{\Theta}}}.\end{aligned} \tag{2.21}$$

The likelihood function $\boldsymbol{p}(\underline{\boldsymbol{y}}\,|\,\underline{\boldsymbol{\Theta}})$ views the observation $\underline{\boldsymbol{y}}$ as a random variable, and views the parameter vector $\underline{\boldsymbol{\Theta}}$ as an unknown constant. Bayes’ Rule in Equation (2.21) represents a method of converting a conditional estimate of the observed data $\underline{\boldsymbol{y}}$ into a conditional estimate of the parameters $\underline{\boldsymbol{\Theta}}$ [7]. This new conditional distribution $\boldsymbol{p}(\underline{\boldsymbol{\Theta}}\,|\,\underline{\boldsymbol{y}})$ is the so-called *posterior distribution*. The posterior distribution views the parameter vector $\underline{\boldsymbol{\Theta}}$ as the random variable, and the observation $\underline{\boldsymbol{y}}$ as fixed.

The posterior distribution can provide a new definition of the ‘best guess’ of the parameter $\underline{\boldsymbol{\Theta}}$. The highest probability of the posterior is known as the *maximum a posteriori* (MAP) estimate [8] (derived in Section 3.3.3)

$$\underline{\boldsymbol{\Theta}}_{MAP} = \arg\max_{\underline{\boldsymbol{\Theta}}} \boldsymbol{p}(\underline{\boldsymbol{\Theta}}\,|\,\underline{\boldsymbol{y}}). \tag{2.22}$$

Notice that in Equation (2.22), the calculation of $\underline{\boldsymbol{\Theta}}_{MAP}$ need not depend on the denominator of Bayes’ Rule. This is because the denominator of the simpler form of Bayes’ Rule in Equation (2.20) does not depend on the argument $\underline{\boldsymbol{\Theta}}$. However, as shown in Chapter 3, the calculation of the denominator will play an important role in estimators other than the MAP estimation. In particular, the denominator is required when we wish to use the posterior distribution $\boldsymbol{p}(\underline{\boldsymbol{\Theta}}\,|\,\underline{\boldsymbol{y}})$ itself instead of the parameter estimate $\underline{\boldsymbol{\Theta}}_{MAP}$.

The denominator of Equation (2.21) is, in general, difficult to calculate. It represents the probability of the data $\underline{\boldsymbol{y}}$ having ever been observed. Closed-form solutions for such integrals often do not exist. For the ubiquitous case of detecting a signal in noise, the likelihood $\boldsymbol{p}(\underline{\boldsymbol{y}}\,|\,\underline{\boldsymbol{\Theta}})$ is the Gaussian function. The denominator becomes an infinite integral of a weighted Gaussian function. This type of integral has a closed-form solution only for a small set of weight functions. We desire maximum flexibility in the weight function (which becomes the prior distribution $\boldsymbol{p}(\underline{\boldsymbol{\Theta}})$). As shown in the next section, the integral in the denominator of Equation (2.21) can be *approximated* with good accuracy.

2.3 Gaussian Quadrature

In the previous section we reviewed the use of Bayes' Rule as a method for finding the MAP estimate of a parameter $\underline{\boldsymbol{\Theta}}$ from observed data $\underline{\boldsymbol{y}}$. This involved calculating a (typically difficult) integral in the denominator of Equation (2.21).

The Gaussian Quadrature is a numerical technique for solving weighted integrals. There are many texts which describe techniques calculating the Gaussian Quadrature. We follow the approach in [9] due to the author's explanation of the rationale behind the use of orthogonal polynomials rather than the calculation of rules themselves.

Consider the integral of the product of two functions $f(x)$ and $w(x)$ over a prescribed (possibly infinite) range $[a, b]$

$$I_f = \int_a^b f(x)w(x)dx. \tag{2.23}$$

In this research, we consider $f(x)$ to be the product $\boldsymbol{p}(\underline{\boldsymbol{y}}|\underline{\boldsymbol{\Theta}})\boldsymbol{p}(\underline{\boldsymbol{\Theta}})$ and will primarily use the unity weight function $w(x) = 1$ with $[a, b] = [-1, 1]$. However, in keeping with [9–11] the general window $w(x)$ is retained.

It is desirable to approximate the integral I_f with a weighted set of samples of the function $f(x)$.

$$I_f \approx \sum_{i=1}^{N} c_i f(x_i) \tag{2.24}$$

The process of approximating an integral as a weighted sum of samples is known as a *quadrature* [11]. The choice of the *abscissa points* $\{x_i\}$ and *weights* $\{c_i\}$ greatly influences the accuracy of the quadrature estimate. It is important to realize that the choice of the $\{x_i\}$ will not depend on the function $f(x)$. In Gaussian quadrature (the type considered here) the user does not specify $\{x_i\}$ at all, but merely the *number* of abscissas, N, that will meet a specified error tolerance. We will derive the optimum set of points $\{x_i\}$ based solely on N. In the description below the $\{x_i\}$ will be left free until the last step when the properties of orthogonal polynomials will be used to define their location.

2.3.1 Polynomial Interpolation.

The primary method of solving Equation (2.23) by quadrature is a two-step process. First, the function $f(x)$ is decomposed into a series of polynomials that approximate the function and whose value is exact at the abscissas

$$f(x) \approx p_f(x) = \sum_{k=0}^{M-1} \sum_{i=1}^{N} \alpha_{k,i} x^k \tag{2.25}$$

$$p_f(x_i) = f(x_i).$$

This process is known as *interpolation*. Next the linearity property of integration is used to integrate each interpolating polynomial separately. If the interpolating polynomials are chosen carefully, the integral of each interpolating polynomial becomes the coefficients $\{c_i\}$ in Equation (2.24).

2.3.1.1 Laplace Interpolation.

It is assumed that the value of $f(x)$ is known everywhere. It is desirable that the polynomial approximation be exact at the abscissa points $\{x_i\}$, although it may not be exact between these points. One method of constructing the appropriate polynomial approximation is by the *Laplace interpolation* ([9], pg. 20).

The Laplace interpolation constructs a set of polynomials $\{l_i(x)\}$ that each has unity value at the i^{th} abscissa x_i and has roots (zeros) at the other $\left\{x_{j \neq i}\right\}$ abscissas. Let $\pi(x)$ be a polynomial whose roots are each abscissa

$$\pi(x) = \prod_{i=1}^{N} (x - x_i). \tag{2.26}$$

A series of polynomials based on Equation (2.26) can be constructed. For each polynomial in the series a single unique root is canceled

$$\pi_j^{\dagger}(x) = \prod_{\substack{i=1 \\ i \neq j}}^{N} (x - x_i) = \frac{\pi(x)}{(x - x_j)}. \tag{2.27}$$

The polynomials $\left\{\pi_j^{\dagger}\right\}$ have the requisite locations of roots, but each must be normalized at the location of its ‘missing’ root. This normalized set of polynomials are the *Laplace factors* $l_i(x)$ and are defined as

$$\begin{aligned} l_i(x) &= \frac{(x-x_1)...(x-x_{i-1})(x-x_{i+1})...(x-x_N)}{(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_N)} \\ &= \prod_{\substack{j=1\\ j\neq i}}^{N} \frac{(x-x_j)}{(x_i-x_j)} \\ &= \frac{\pi(x)}{(x-x_i)\pi_i^{\dagger}(x_i)}. \end{aligned} \tag{2.28}$$

Because each Laplace factor has unity value at its ‘own’ root, the interpolating polynomial $p_f(x)$ is the weighted sum of the $\{l_i(x)\}$,

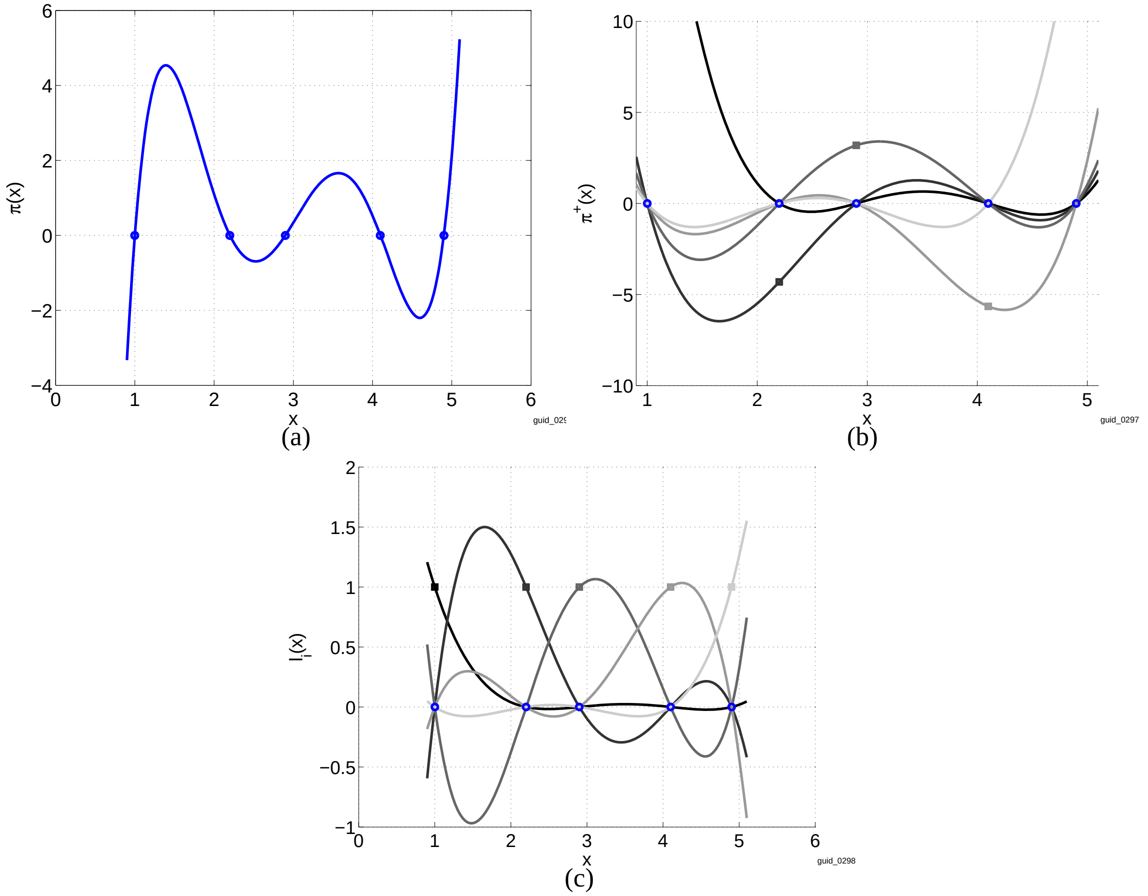
$$f(x) \approx p_f(x) = \sum_{i=1}^{N} f(x_i) l_i(x). \tag{2.29}$$

To help clarify the above discussion, the development of the Laplace interpolation for a function with five abscissas is shown in Figure 2.2. In Figure 2.2(a), the function $\pi(x)$ is composed of five roots at the circled locations. Figure 2.2(b) shows the development of the five ‘missing root’ polynomials $\pi_i^{\dagger}(x)$. Notice that only one of the polynomials is nonzero at each abscissa (as marked by the square). Finally, Figure 2.2(c) shows the Laplace factors $l_i(x)$. Note that each polynomial now has unity value at its corresponding abscissa (again shown as squares).

At this point, the actual values of the abscissas are still unspecified. Soon it will be shown that specifying the polynomial $\pi(x)$ directly will lead to desirable results.

2.3.1.2 Error Term.

In order to calculate the interpolation error $E(x) = f(x) - p_f(x)$ we make use of the Rolle Theorem ([9], pg 20) which states that for any continuous function $f(x)$ with roots a and b there exists a point $a < \xi < b$ where its derivative $f'(\xi) = 0$. Equivalently, for any smooth function there is at least one local minimum or maximum between the zeros of the function. Therefore for a function $f(x)$ with a set of N roots $\{x_i\}$, its derivative $f'(x)$ will have $N - 1$ roots located *inside* $\min\{x_i\} < x < \max\{x_i\}$. Equivalently for the polynomial



(a) (b) (c)

Figure 2.2: Laplace polynomial development: (a) Product of abscissas $\pi(x)$, (b) Polynomials $\pi_i^{\dagger}(x)$ created by removing 1 root each, and (c) Laplace factors $l_i(x)$.

$p_f(x)$ in Equation (2.29), whose roots are the N abscissa points, there will be $N - 1$ points where the derivative $p'_f(x) = 0$.

Let the auxiliary function $F(x)$ [9, 12, 13] be

$$\begin{aligned} F(x) &= E(x) - K\pi(x) \\ &= f(x) - p_f(x) - K\pi(x), \end{aligned} \tag{2.30}$$

which has N roots from the term $f(x) - p_f(x)$. The factor K will be chosen to give an additional zero at some new point $\hat{x}$. The first derivative $F'(x)$ has N roots, the second derivative $F''(x)$ has $N - 1$ roots and so on. The N^{th} derivative $F^{(N)}(x)$ has one root, ξ, and by Rolle's Theorem it is on the interior of the abscissas $\{x_i\}$. As a root, Equation (2.30) becomes

$$\begin{aligned} F(\xi) &= 0 \\ &= f^{(N)}(\xi) - p_f^{(N)}(\xi) - K\pi^{(N)}(\xi). \end{aligned} \tag{2.31}$$

Note that the polynomial interpolation $p_f(x)$, being a linear combination of each $l_i(x)$, is of order $N-1$, and so its N^{th} derivative is zero. The function $\pi(x)$ is order N and monic (leading coefficient is unity), therefore $\pi^{(N)}(\xi) = N!$. Equation (2.31) becomes

$$\begin{aligned} f^{(N)}(\xi) &= KN! \\ K &= \frac{1}{N!} f^{(N)}(\xi). \end{aligned} \tag{2.32}$$

Recall that ξ is the point where $F^{(N)}(x) = 0$. The error in the interpolation is

$$\begin{aligned} E(x) &= f(x) - p_f(x) \\ &= F(\xi) + K\pi(x) \qquad \text{(from Equation (2.30))} \\ &= K\pi(x) \\ &= \frac{1}{N!} f^{(N)}(\xi)\pi(x). \end{aligned} \tag{2.33}$$

Equation (2.32) shows that the interpolation error goes inverse-factorial with the number of abscissas, and has a shape that follows a polynomial whose roots are the abscissa points $\{x_i\}$. The worst-case error can be conservatively estimated to be of amplitude

$$E_{max} = \frac{\pi(x)}{N!}\left(\max\left|f^{(N)}(\xi)\right|\right) \quad \text{over the range} \quad [\min\{x_i\}, \max\{x_i\}]. \tag{2.34}$$

Note that for any polynomial function $f_M(x)$ of order $M < N-1$, the derivative term $f_M^{(N)}(\xi) = 0$, and therefore the polynomial interpolation $p_f(x)$ has zero error inside the abscissas.

2.3.2 Hermite Interpolation.

The Laplace interpolation, Equation (2.29), is exact for polynomials up to order $N-1$. Following ([9], sec 2.6), the Hermite interpolation further extends the Laplace interpolation by requiring the interpolating polynomial $p_f(x)$ *and* its first derivative to be exact at the abscissas. The Hermite interpolation defines two new polynomials $h_i(x)$ and $\overline{h_i}(x)$ to interpolate the function $f(x)$ *and* its derivative $f'(x)$. This is an improvement over Laplace interpolation since the added constraints on the derivative allow us to specify a higher order interpolating polynomial $p_f(x)$ with the same number of abscissas. Under these conditions, a new interpolating polynomial $p_f(x)$ is defined as

$$p_f(x) = \sum_{i=1}^{N} h_i(x) f(x_i) + \sum_{i=1}^{N} \overline{h_i}(x) f'(x_i). \tag{2.35}$$

It follows that the derivative of Equation (2.35) is

$$\begin{aligned} p_f'(x) &= \frac{d}{dx}\left(\sum_{i=1}^{N} h_i(x) f(x_i)\right) + \frac{d}{dx}\left(\sum_{i=1}^{N} \overline{h_i}(x) f'(x_i)\right) \\ &= \sum_{i=1}^{N} \frac{d}{dx}\Big(h_i(x) f(x_i)\Big) + \sum_{i=1}^{N} \frac{d}{dx}\Big(\overline{h_i}(x) f'(x_i)\Big) \\ &= \sum_{i=1}^{N}\Big(h_i'(x) f(x_i) + h_i(x) f'(x_i)\Big) + \sum_{i=1}^{N}\Big(\overline{h_i}'(x) f'(x_i) + \overline{h_i}(x) f''(x_i)\Big). \end{aligned} \tag{2.36}$$

Exact reconstruction at the abscissas requires the values of $h_i(x)$ be unity at the i^{th} abscissa and zero at the other abscissas while its derivative should be zero at all abscissas. Similarly, the derivative of the polynomial $\overline{h_i}(x)$ should be unity at the i^{th} abscissa and zero at the other abscissas while the value of $\overline{h_i}(x)$ should be zero at all abscissas. It is convenient to make use of the Kronecker delta δ_{ij} and so the conditions can be restated as

$$\begin{array}{rclcrcl} h_i(x_j) & = & \delta_{ij} & \qquad & \overline{h_i}(x_i) & = & 0 \\ h_i'(x_i) & = & 0 & & \overline{h_i}'(x_j) & = & \delta_{ij}. \end{array} \tag{2.37}$$

The conditions in Equation (2.37) force the value of $p_f(x)$ and its derivative to be

$$p_f(x_i) = f(x_i) \tag{2.38}$$

$$p_f'(x_i) = 2f'(x_i). \tag{2.39}$$

For N abscissa, the conditions on values and derivatives in Equation (2.37) are sufficient to define the coefficients of a polynomial $p_f(x)$ of order $2N-1$. The Laplace factor defined in Equation (2.28) is used as a template for $h_i(x)$ and $\overline{h_i}(x)$. Squaring the Laplace factor $l_i(x)$ yields a polynomial of order $2(N-1)$. By applying leading factors $r_i(x)$ and $s_i(x)$ to $[l_i(x)]^2$, two unique polynomials of order $2N-1$ are created. In order to satisfy Equation (2.37), [9] suggests the following values for $r_i(x)$ and $s_i(x)$,

$$\begin{aligned} h_i(x) &= r_i(x)[l_i(x)]^2, \\ &= [1 - 2l_i'(x_i)(x - x_i)][l_i(x)]^2, \end{aligned} \tag{2.40}$$

$$\begin{aligned} \overline{h_i}(x) &= s_i(x)[l_i(x)]^2, \\ &= (x - x_i)[l_i(x)]^2. \end{aligned} \tag{2.41}$$

The subsequent error term is derived similarly to Equation (2.32) and is given in [9] as

$$\begin{aligned} E(x) &= f(x) - p_f(x) \\ &= \frac{f^{(2N)}(\xi)}{(2N)!}[\pi(x)]^2. \end{aligned} \tag{2.42}$$

Note that for any function $f(x)$ that is a polynomial of order $2N-1$ or lower, $f^{(2N)} = 0$ and therefore the interpolation error is zero (exact reconstruction).

2.3.3 Integration and Integration Error.

To solve for the integral Equation (2.23), integrate the terms in Equation (2.35).

$$\begin{aligned}
I_f &= \int_a^b f(x)w(x)dx \approx I_{p_f} \\
I_{p_f} &= \int_a^b \left(\sum_{i=1}^{N} h_i(x) f(x_i) + \sum_{i=1}^{N} \overline{h_i}(x) f'(x_i) \right) w(x)dx \\
&= \sum_{i=1}^{N} \left(f(x_i) \underbrace{\int_a^b h_i(x)w(x)dx}_{H_i} \right) + \sum_{i=1}^{N} \left(f'(x_i) \underbrace{\int_a^b \overline{h_i}(x)w(x)dx}_{\overline{H_i}} \right) \\
&= \sum_{i=1}^{N} H_i f(x_i) + \sum_{i=1}^{N} \overline{H_i} f'(x_i). \qquad (2.43)
\end{aligned}$$

Note that the inner integrals, which have been labeled H_i and $\overline{H_i}$ depend only on the weight function $w(x)$ and (via the definitions of $h_i(x)$ and $\overline{h_i}(x)$) the location of the abscissas, but not on the function $f(x)$ that we are trying to interpolate. This allows H_i and $\overline{H_i}$ to be pre-computed.

The error term given in Equation (2.42) gives the interpolation error. To get the integrated error E, simply integrate Equation (2.42)

$$\begin{aligned}
E &= \int_a^b f(x)w(x)dx - \int_a^b p_f(x)w(x)dx \\
&= \int_a^b E(x)w(x)dx \\
&= \int_a^b \frac{f^{(2N)}(\xi)}{(2N)!} [\pi(x)]^2 w(x)dx \\
&= \frac{f^{(2N)}(\eta)}{(2N)!} \int_a^b [\pi(x)]^2 w(x)dx. \qquad (2.44)
\end{aligned}$$

The last part of Equation (2.44) makes use of the Mean Value Theorem ([14], pg 84),

$$\int_a^b f(x)g(x)dx = f(\hat{x}) \int_a^b g(x)dx, \quad a \le \hat{x} \le b. \qquad (2.45)$$

A new value η in the interval $[\min\{x_i\}, \max\{x_i\}]$ sets the amplitude of the error. The actual value of η is not important for the error analysis, rather the worst case error is the largest value of $f^{(2N)}(x)$ in the interval defined by smallest and largest roots. The error decreases with the number of terms as $1/(2N)!$. Finally, the scale is determined by the integral of Equation (2.44). This factor is solely dependent on the choice of abscissa points which define $\pi(x)$ and the weight function $w(x)$.

2.3.4 Orthogonal Polynomials.

The Hermite interpolation quadrature is exact for solving integrals of order $2N - 1$ and lower. Unfortunately, as written in Equation (2.43) it requires the evaluation of $2N$ coefficients and the calculation of the derivatives $f'(x)$. Recall the definition of the derivative coefficients from Equation (2.43)

$$\begin{aligned}\overline{H_i} &= \int_a^b \overline{h_i}(x)w(x)dx \\ &= \int_a^b (x - x_i)[l_i(x)]^2 w(x)dx \\ &= \int_a^b (x - x_i)\frac{[\pi(x)]^2}{(x - x_i)^2[\pi_i^\dagger(x_i)]^2} w(x)dx \\ &= \int_a^b \frac{\pi(x)}{\pi_i^\dagger(x_i)}\frac{\pi(x)}{(x - x_i)\pi_i^\dagger(x_i)} w(x)dx \\ &= \frac{1}{\pi_i^\dagger(x_i)}\int_a^b \pi(x)l_i(x)w(x)dx. \end{aligned} \tag{2.46}$$

The coefficients $\overline{H_i}$ become zero if the integral in Equation (2.46) is zero. This is equivalent to the two polynomials $\pi(x)$ and $l_i(x)$ being *orthogonal* under the weight $w(x)$. Quadrature rules of the form of Equation (2.24) with orthogonal polynomials are called *Gaussian Quadratures* [9][11]. These quadratures are very efficient due to their high-degree polynomial accuracy and low number of nonzero coefficients.

The orthogonality can be expressed as an inner product in a Hilbert space defined by $w(x)$

$$\langle \pi, l_i \rangle_w = \int_a^b \pi(x)l_i(x)w(x)dx. \tag{2.47}$$

Two polynomials $p_i(x)$ and $p_j(x)$ are orthogonal if $\left\langle p_i, p_j \right\rangle = 0$ for $i \neq j$.

The polynomial $\pi(x)$ is composed of roots at the abscissas, and is of order $N - 1$. Evans [9] notes that for $\pi(x)$ to be orthogonal to all $l_i(x)$ it is sufficient to be orthogonal to *all* polynomials of order $N - 2$ and less. This is accomplished by ensuring all $\pi(x)$ is orthogonal to all other polynomials.

The orthogonality condition sets the value of $\overline{H_i}$ to zero, and the integrated Hermite interpolation in Equation (2.43) becomes

$$\begin{aligned} I_f &= \int_a^b f(x)w(x)dx \approx I_{p_f} \\ I_{p_f} &= \sum_{i=1}^{N} H_i f(x_i) \end{aligned} \tag{2.48}$$

Note that H_i in Equation (2.48) is the same form as c_i in Equation (2.24). This leads to the definition of the weight coefficients as

$$\begin{aligned} c_i = H_i &= \int_a^b h_i(x)w(x)dx \\ &= \int_a^b [1 - 2l_i'(x_i)(x - x_i)][l_i(x)]^2 w(x)dx. \end{aligned} \tag{2.49}$$

The orthonormal polynomials for a given weight $w(x)$ are unique. Techniques for determining the coefficients are given in [9–11, 15]. In general the polynomial coefficients are irrational, but extensive tables of numerical values are given in [16].

2.3.5 Practical Use.

The above derivation shows that integration under a specific weighting function, $w(x)$, yields a set of orthonormal polynomials which are unity at their roots, and those roots become the abscissa points for the quadrature. It is convention to name the set of coefficients c_i and abscissas $\{x_i\}$ after the underlying orthogonal polynomial. In practice several standard orthogonal polynomials are used almost exclusively and these are summarized in Table 2.1.

Many of the orthogonal polynomials are only orthogonal over ‘standard’ bounds $[-1, 1]$. If the problem requires integration of bounds $[a, b]$, the *affine transformation* [9]

Table 2.1: Common Gaussian Quadrature Rules (after [10]).

Polynomial Name	$\mathbf{w(x)}$	$[\mathbf{a}, \mathbf{b}]$
Legendre	1	$[-1, 1]$
Hermite	e^{-t^2}	$[-\infty, \infty]$
Laguerre	e^{-t}	$[0, \infty]$
Chebyshev (first kind)	$\frac{1}{\sqrt{1-x^2}}$	$[-1, 1]$
Chebyshev (second kind)	$\sqrt{1-x^2}$	$[-1, 1]$

may be used to transform the abscissas $\{x_i\}$, which integrate over the standard bounds, to new abscissas x'_i which integrate over $[a, b]$:

$$x' = \frac{b-a}{2}x + \frac{b+a}{2} \tag{2.50}$$

$$\int_{-1}^{1} f(x)w(x)dx = \int_{a}^{b} f(x')w(x')dx'.$$

Under this transformation the quadrature weights c_i do not change. The new quadrature rule is

$$x'_i = \frac{b-a}{2}x_i + \frac{b+a}{2} \tag{2.51}$$

$$\int_{a}^{b} f(x)w(x)dx \approx \sum_{i=1}^{N} f(x'_i)c_i. \tag{2.52}$$

Based on the use of tables of pre-computed polynomials, the following steps can be used to approximately compute the integral of $f(x)$:

1. Based on the weight function $w(x)$ for the problem at hand, decide which orthogonal polynomial is most appropriate.

2. Decide on a polynomial order N (the number of abscissa points).

3. Compute (or look up in a table) the roots of the N^{th}-order orthogonal polynomial. These will be the abscissa points $\{x_i\}$.

4. Compute (or look up in a table) the weight coefficients. These will be the c_i.

5. Compute the value of the function at the N abscissa points (transform if necessary).

6. Multiply each weight c_i with the function value $f(x_i)$.

7. Sum these terms together. The result is the approximation of the integral.

2.3.6 Summary of Gaussian Quadrature.

We now recall the important points of the derivation of the Gaussian quadrature:

- It is desired to approximate the weighted integral of a function $f(x)$ as a sum of N abscissa points.

- The function is approximated by a polynomial $p_f(x)$, and the weighted integral of $p_f(x)$ is an approximation of the weighted integral of $f(x)$.

- The first approximation (the Laplace interpolation) of $f(x)$ is the weighted sum of Laplace factors $l_i(x)$. The Laplace factors are polynomials constructed so that each $l_i(x)$ has unity value at a specific abscissa and zero value at the other abscissas. The weight coefficients are the values of $f(x)$ at the abscissas.

- The error in the Laplace interpolation approximation of $f(x)$ is proportional to the N^{th} derivative of $f(x)$, and scales inversely with $N!$.

- An improved approximation (the Hermite interpolation) is a weighted sum of two polynomials $h_i(x)$ and $\overline{h_i}(x)$. The coefficients of $h_i(x)$ are the values of $f(x)$ at the abscissas and the coefficients of $\overline{h_i}(x)$ are the values of the derivative $f'(x)$ at the abscissas. The polynomials $h_i(x)$ and $\overline{h_i}(x)$ are each constructed from the Laplace factors.

- The error in the Hermite interpolation approximation of $f(x)$ is proportional to the $(2N)^{th}$ derivative of $f(x)$, and scales inversely with $(2N)!$.

- The integral of $f(x)$ is approximately the integral of the Hermite interpolation polynomial $p_f(x)$. The weighted integrals of $h_i(x)$ and $\overline{h_i}(x)$ are independent of $f(x)$ and may be pre-computed.

- If the abscissas correspond to the roots of an orthonormal polynomial, the derivative term in the Hermite interpolation $p_f(x)$ integrates to zero. Therefore only the N computations of the values of $f(x)$ at the abscissa points are necessary to achieve the Hermite interpolation error performance.

In the next section, we show how the Gaussian quadrature can be used to solve integrals like those found calculating the posterior distribution.

2.4 Calculation of Bayesian Posterior

The calculations described in Equation (2.21) involve the calculation of the observation probability $\boldsymbol{p}(\underline{\boldsymbol{y}})$. In general, this is accomplished by computing a difficult (sometimes impossible) integral. The denominator of Equation (2.21) serves to ensure that the posterior distribution is a true probability distribution, i.e. it integrates to one. For many problems, [8, 17] the MAP estimate given in Equation (2.22) is merely a search for the highest value in function space and the actual value of the peak probability density is unimportant.

For this research, we will often compare results of different probability spaces and the normalization constant that is the denominator of Equation (2.21) will not be the same in each space. Further, in the measurement of confidence described below, it will be seen that integrals of this sort over finite intervals have meaning as probability masses. It will therefore be necessary to calculate this *D*-dimensional integral (repeated here for convenience)

$$\boldsymbol{p}(\underline{\boldsymbol{y}}) = \int_{\mathbb{R}^D} \boldsymbol{p}(\underline{\boldsymbol{y}} \,|\, \underline{\boldsymbol{\Theta}})\boldsymbol{p}(\underline{\boldsymbol{\Theta}})d\underline{\boldsymbol{\Theta}}. \tag{2.53}$$

2.4.1 Monte-Carlo Numerical Integration.

The Monte-Carlo method is a sampling-based method. It can be viewed as an application to the simplest estimator for the expected value a random variable X: the average of a large number of samples. Consider the expected value [6] of a function $f(x)$ of some continous random variable X

$$\boldsymbol{E}[f(x)] = \int_{-\infty}^{\infty} f(x)\boldsymbol{p}(x)dx. \tag{2.54}$$

The integral in Equation (2.54) can be approximated using the Strong Law of Large Numbers [18] as

$$\boldsymbol{E}[f(x)] = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} f(x_i), \qquad x_i \sim \boldsymbol{p}(X). \tag{2.55}$$

For integrals of the form of Equation (2.53) we choose the evaluated function $f(x) = \boldsymbol{p}(\underline{\boldsymbol{y}} \,|\, \underline{\boldsymbol{\Theta}})$ and the weight probability as $\boldsymbol{p}(\underline{\boldsymbol{\Theta}})$. The integral then becomes

$$\begin{aligned} \int_{\mathbb{R}^D} \boldsymbol{p}(\underline{\boldsymbol{y}} \,|\, \underline{\boldsymbol{\Theta}})\boldsymbol{p}(\underline{\boldsymbol{\Theta}})d\underline{\boldsymbol{\Theta}} &\approx \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{p}(\underline{\boldsymbol{y}} \,|\, \underline{\boldsymbol{\Theta}}_i), \qquad \underline{\boldsymbol{\Theta}}_i \sim \boldsymbol{p}(\underline{\boldsymbol{\Theta}}) \\ &= \hat{\boldsymbol{p}}(\underline{\boldsymbol{y}}). \end{aligned} \tag{2.56}$$

Note that the integrated function $\boldsymbol{p}(\underline{\boldsymbol{y}} \mid \underline{\boldsymbol{\Theta}}_i)$ is dependent on the observed data $\underline{\boldsymbol{y}}$. This estimate of the denominator requires repeated draws of a multidimensional $\underline{\boldsymbol{\Theta}}_i$ from a random number generator with a precise distribution. Several methods, such as the Metropolis sampler, and Gibbs sampler [17] can be used to generate the required random draws.

Because the Monte Carlo process can be viewed as an estimator of $\boldsymbol{p}(\underline{\boldsymbol{y}})$ it has well-studied properties. Although Monte Carlo integration is a commonly-used technique it is recommended ([19], pg. 162) for low-accuracy integrals of functions that are not strongly peaked. As we will show in Chapter 4, often the function we wish to integrate (the likelihood function) is Gaussian and can be quite peaked. Therefore, we will not use Monte Carlo integration further in this research.

2.4.2 Gauss-Legendre Quadrature Numerical Integration.

The results of the Gaussian quadrature developed in Section 2.3 can be applied to integrals of the form of Equation (2.53). Recall that the Gaussian quadrature solves integrals of the form

$$\int_a^b f(x)w(x)dx \approx \sum_i^N c_i f(x_i).$$

All of the integrals of interest in this research will be of finite bounds (usually due to limitations on the priors, see Section 3.2). The choice of weight function will, in general, be the uniform function $w(x) = 1$. Based on the summary in Table 2.1, these two constraints lead to the use of the Gauss-Legendre rules. We will make use of the affine transformation described in Section 2.3.

This research focuses on parameter sets $\underline{\boldsymbol{\Theta}}$ that are multidimensional. Let each scalar parameter θ_i be bounded by a corresponding region in parameter space $\mathcal{P}_i$. For a given number D of independent parameters, the integral expands as follows

$$\begin{aligned}\boldsymbol{p}(\underline{\boldsymbol{y}}) &= \int_{\mathbb{R}^D} \boldsymbol{p}(\underline{\boldsymbol{y}} \,|\, \underline{\boldsymbol{\Theta}})\boldsymbol{p}(\underline{\boldsymbol{\Theta}})d\underline{\boldsymbol{\Theta}} \\ &= \int_{\mathcal{P}_1}\int_{\mathcal{P}_2} \cdots \int_{\mathcal{P}_D} \boldsymbol{p}(\underline{\boldsymbol{y}} \,|\, \theta_1, \theta_2, ...\theta_D)\boldsymbol{p}(\theta_1, \theta_2, ...\theta_M)\ d\theta_D...d\theta_2...d\theta_1. \end{aligned} \tag{2.57}$$

The Gauss-Legendre quadrature is only defined for integrals over the range $[-1, 1]$. The calculation of Equation (2.53) requires integration limits $[\min\theta, \max\theta]$. The affine transformation of the Legendre roots $\{x_i\}$ yields new abscissa points for the required integration limits ([9], pg 42)

$$\theta_{k_i} = \left(\frac{\max\mathcal{P}_k - \min\mathcal{P}_k}{2}\right)x_i + \left(\frac{\max\mathcal{P}_k + \min\mathcal{P}_k}{2}\right). \tag{2.58}$$

Each integral of Equation (2.57) can be evaluated as a separate Gauss-Legendre quadrature, under the affine transformation. For the Bayes integrand $\boldsymbol{p}(\underline{\boldsymbol{y}} \,|\, \underline{\boldsymbol{\Theta}})\boldsymbol{p}(\underline{\boldsymbol{\Theta}})$ that can be approximated by a quadrature of order N, the denominator (Equation (2.57)) becomes

$$\begin{aligned}\boldsymbol{p}(\underline{\boldsymbol{y}}) &\approx \sum_{k_1=1}^{N} c_{k_1} \sum_{k_2=1}^{N} c_{k_2} \cdots \sum_{k_D=1}^{N} c_{k_D} \boldsymbol{p}(\underline{\boldsymbol{y}} \,|\, \theta_{k_1}, \theta_{k_2}...\theta_{k_D})\boldsymbol{p}(\theta_{k_1}, \theta_{k_2}...\theta_{k_D}) \\ &\approx \sum_{k_1=1}^{N}\sum_{k_2=1}^{N} \cdots \sum_{k_D=1}^{N} (c_{k_1}c_{k_2}\ ...\ c_{k_D})\ \boldsymbol{p}(\underline{\boldsymbol{y}} \,|\, \theta_{k_1}, \theta_{k_2}...\theta_{k_D})\boldsymbol{p}(\theta_{k_1}, \theta_{k_2}...\theta_{k_M}). \end{aligned} \tag{2.59}$$

Thus, the integral Equation (2.53) can be approximated by a weighted sum of abscissa points. The coefficients $\{c_{k_i}\}$ are defined by the Legendre polynomials and available in numerical form from references such as [15] and [16]. These integral solutions enable the calculation of the posterior density $\boldsymbol{p}(\underline{\boldsymbol{\Theta}} \,|\, \underline{\boldsymbol{y}})$, which is then used to find optimum estimates of the parameters.

2.5 Credibility and Confidence

Typical parameter estimation methods, such as maximum likelihood (Equation (2.19)) and maximum a posteriori (Equation (2.22)), aim to find a single parameter vector $\underline{\boldsymbol{\Theta}}$ for the given data $\underline{\boldsymbol{y}}$. However, often the consumer of the estimate (such as higher-level ATR software) desires more than just the best estimate; they may desire some measure of how 'good' the estimate is. In the statistics community, it is common to distinguish between these measures of goodness for Bayesian methods like the MAP estimate of Equation (2.22) and the 'likelihoodist' or 'frequentist' methods like the MLE estimate of Equation (2.19). The mathematical techniques are very similar. This research focuses on Bayesian methods and so the next section considers the idea of the Credible Interval: a Bayesian idea similar to the frequentist 'Confidence Interval'.

2.5.1 Credible Interval.

The 'credible' interval is used for Bayesian measurements as a measure of goodness-of-estimate. Gill, [17] p.45, defines a credible interval as a contiguous region C in parameter space such that a certain posterior probability mass α is attained:

$$C = \arg_{C'} \left\{ \int_{C'} \boldsymbol{p}(\underline{\boldsymbol{\Theta}} \,|\, \underline{\boldsymbol{y}}) d\underline{\boldsymbol{\Theta}} = \alpha \right\}. \tag{2.60}$$

2.5.2 Intuitive Meaning of the Credible Interval.

Gill [17] poses a very important intuitive definition for the meaning of the credible interval. The credible interval for the desired mass α is the region where the true value of the parameter $\underline{\boldsymbol{\Theta}}$ is covered (100α) percent of the time. The important term is the *coverage* of the true parameter, upon a certain number of repeated observations. Using this definition makes the credible interval similar in sentiment to the idea of 'parameter resolution' meaning the precision of the estimate is controlled by the size of the interval.

Using this definition, the target probability mass α takes on a new meaning. The total probability over all parameter space $\mathcal{P}$ must, by definition, be one

$$\int_{\mathcal{P}} \boldsymbol{p}(\underline{\boldsymbol{\Theta}} \,|\, \underline{\boldsymbol{y}}) d\underline{\boldsymbol{\Theta}} = 1. \tag{2.61}$$

The credible interval C and its compliment $\neg C$ are disjoint and span the whole parameter space

$$C \bigcap \neg C = \emptyset, \\ C \bigcup \neg C = \mathcal{P}. \tag{2.62}$$

By Equation (2.62) the regions C and $\neg C$ form a partition of $\mathcal{P}$, and therefore

$$\int_{\mathcal{P}} \boldsymbol{p}(\underline{\boldsymbol{\Theta}} \,|\, \underline{y}) d\underline{\boldsymbol{\Theta}} = \int_{C} \boldsymbol{p}(\underline{\boldsymbol{\Theta}} \,|\, \underline{y}) d\underline{\boldsymbol{\Theta}} + \int_{\neg C} \boldsymbol{p}(\underline{\boldsymbol{\Theta}} \,|\, \underline{y}) d\underline{\boldsymbol{\Theta}} \\ \int_{\neg C} \boldsymbol{p}(\underline{\boldsymbol{\Theta}} \,|\, \underline{y}) d\underline{\boldsymbol{\Theta}} = 1 - \alpha. \tag{2.63}$$

The Gill definition, when applied to Equation (2.63) leads to the interpretation that the quanitity $1 - \alpha$ represents the probability that the true parameter $\underline{\boldsymbol{\Theta}}$ lies *outside* the credible region. Note that the value of $(1 - \alpha)$ is also a probability and represents the *probability of a miss* [20].

2.6 Computational Requirements

This research will focus on the Gauss-Legendre quadrature as its main technique for solving integrals. As shown in Equation (2.59), for an integral of D parameter dimensions using an N order quadrature, the number of calculations of $\boldsymbol{p}(\underline{\boldsymbol{\Theta}} \,|\, \underline{\boldsymbol{y}})$ scales like $\mathcal{O}(N^D)$.

The received phase history is a complex-valued vector consisting of F frequencies and K azimuth and elevation points. The complex vector is of length FK. The complex phase results from the range terms R_t, R_r and $\Delta R(\underline{\boldsymbol{\Theta}})$ in Equation (2.1). In this research, we consider the received phase history $\underline{\boldsymbol{y}}$ to be the complex phase history 'flattened' into a real vector of length $M = 2FK$. The flattened vector (see [21]) is the concatenation of the real and imaginary parts of the complex vector $\underline{\boldsymbol{S}}_{true}$ given by Equation (2.1) for the given shape and corrupted by noise $\underline{\boldsymbol{N}}$ as

$$\underline{\boldsymbol{y}} = \begin{bmatrix} \Re(\underline{\boldsymbol{S}}_{true}) \\ \Im(\underline{\boldsymbol{S}}_{true}) \end{bmatrix} + \underline{\boldsymbol{N}}. \tag{2.64}$$

For consistency we will make use of a similarly flattened parametrized test phase history $\underline{\boldsymbol{S}}_\Gamma$ as

$$\underline{\boldsymbol{S}}_\Gamma = \begin{bmatrix} \mathfrak{Re}(\underline{\boldsymbol{S}}(k)) \\ \mathfrak{Im}(\underline{\boldsymbol{S}}(k)) \end{bmatrix}. \tag{2.65}$$

The subscript Γ represents the shape type and thus the particular form of the model $\underline{\boldsymbol{S}}_\Gamma$ (see [1, 21]).

The likelihood density $\boldsymbol{p}(\underline{\boldsymbol{y}}|\underline{\boldsymbol{\Theta}})$ used in this research represent received shape phase histories in the presence of Additive White Gaussian Noise (AWGN). The resulting likelihood is normally distributed whose mean is the (flattened) Jackson phase history $\underline{\boldsymbol{S}}_\Gamma(\underline{\boldsymbol{\Theta}})$ and covariance $\sigma^2\underline{\underline{\boldsymbol{I}}}$ ([18])

$$\boldsymbol{p}(\underline{\boldsymbol{y}}|\underline{\boldsymbol{\Theta}}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^M \exp\left[-\frac{1}{2\sigma^2}\left(\underline{\boldsymbol{y}} - \underline{\boldsymbol{S}}_\Gamma(\underline{\boldsymbol{\Theta}})\right)^T\left(\underline{\boldsymbol{y}} - \underline{\boldsymbol{S}}_\Gamma(\underline{\boldsymbol{\Theta}})\right)\right]. \tag{2.66}$$

The inner product

$$\left(\underline{\boldsymbol{y}} - \underline{\boldsymbol{S}}_\Gamma(\underline{\boldsymbol{\Theta}})\right)^T\left(\underline{\boldsymbol{y}} - \underline{\boldsymbol{S}}_\Gamma(\underline{\boldsymbol{\Theta}})\right) \tag{2.67}$$

is a computationally intensive operation. For an observation vector $\underline{\boldsymbol{y}}$ of length $2FK$ there are $2FK$ multiply-and-add operations, as well as $2FK$ subtractions. The resulting complexity is $\mathcal{O}(2FKN^D)$ operations.

This results in an enormous computational burden on the simulation platform. In order to efficiently perform the Gauss-Legendre quadrature, an efficient implementation of Equation (2.59) is required.

2.6.1 Graphics Processor Units.

A calculation of the form of Equation (2.59) is inherently bound by the speed of the simulation platform. In a standard microprocessor system each term of each summand is evaluated in a lockstep fashion as a `for` loop. Prior to the late 1990s microprocessor researchers focused their efforts on making the processor itself faster. The result was high Gigahertz clock speeds. However the demands for ever increasing computation power, particularly in real-time graphics and gaming [22], began to outpace microprocessor speeds. The result was a shift at the start of the 21st century towards graphics cards with multiple, simple processor cores designed for the type of repetitive numeric computations used in graphics simulation. These specialized graphics processors became known as Graphics Processing Units (GPUs). Unlike CPU microprocessors which may have 4 to 8 cores, a typical GPU can have hundreds or thousands of microprocessors running simultaneously.

2.6.2 CUDA GPUs.

Early GPUs were developed specifically for graphics-intensive operations like OpenGL and DirectX [22]. The CUDA platform was developed by the NVIDIA Corporation to allow programmers access to the computational capabilities of the graphics card for general-purpose and scientific computing. The CUDA programming interface is increasingly supported in the MATLAB programming environment. Over the course of this research MATLAB was used to define the architectural framework and high-level computation, and calls to the CUDA processors were used for the intense repetitive computations.

Developing CUDA software requires writing code for specialized compilers provided by NVIDIA and Microsoft [22, 23]. The CUDA language itself is an extended version of C++. The programmer writes the algorithm implementation similarly to a standard C++ program which is compiled into a pseudo-assembly language called a PTX file. These files are portable across different CUDA processors. The GPU has its own memory (with some exceptions, see [22]) that is independent of the main computer memory. The CUDA features used in this research cannot use computer main memory, nor can computer applications use GPU memory directly. The data must be explicitly transferred, and this can be a considerable bottleneck.

Execution of the PTX file invokes the CUDA driver, a Windows- or Linux device driver that interfaces with the GPU hardware. The driver loads the PTX file onto the GPU, executes the code, transfers the relevant data, then returns control of the GPU back to the operating system for normal graphics use (such as updating the screen).

2.6.3 MATLAB Use.

When a user wishes to execute an algorithm on the GPU via MATLAB a special MATLAB object called a *CUDA Kernel* is created with the location of the PTX file given. At this point the CUDA driver pre-loads the PTX file into the GPU. The kernel can be executed like a normal MATLAB function.

As noted before, memory transfers from the computer main memory (where MATLAB can use it for, e.g., drawing figures) and the GPU memory (where the GPU uses it) must be done carefully. MATLAB virtualizes the objects (such as matrices and vectors) held on the GPU in the form of the `gpuArray` data type. Although in most cases MATLAB will transparently perform the transfer, it is advisable to keep the data on the GPU for as long as possible. Most MATLAB functions such as arithmetic, trigonometry and memory management can be performed directly on the GPU. The user may then manually transfer back to 'normal' MATLAB memory for graphics plotting or display by use of the `gather` function.

2.7 Summary

This chapter reviewed the Jackson phase history models for six canonical shapes. The idea of a 'best guess' estimation of the parameter vector $\underline{\boldsymbol{\Theta}}$ requires the use of elements from probability theory. The likelihood function $\boldsymbol{p}(\underline{\boldsymbol{y}}\,|\,\underline{\boldsymbol{\Theta}})$ views the observation $\underline{\boldsymbol{y}}$ as random and the parameter is fixed but unknown. Bayes' Rule is used to incorporate prior knowledge about the parameters and changes the view to that of a random parameter $\underline{\boldsymbol{\Theta}}$ and a fixed observation $\underline{\boldsymbol{y}}$ which is known. Bayes' Rule requires solving of a difficult integral which is aided by the Gaussian quadrature. The idea of confidence in the estimate of the parameter was introduced as the credible interval. The credible interval allows us to make statements the probability that the true parameter vector is located in a region of possible parameter guesses. Finally, the CUDA GPU was introduced and we describe how it is an enabling technology to perform the large number of calculations needed to solve the integration problem of Bayes' Rule.

III. Methodology

This section describes new techniques and the methodology developed over the course of this research. The assumptions and methods described herein will carry to Chapter 4.

3.1 Assumptions and Data Model

3.1.1 Radar Environment.

This research assumes the received data is in the form of complex (I and Q) phase history. The radar system is assumed to be a monostatic SAR operating in spotlight mode, however the technique can be extended to bistatic radar supported by the Jackson model. The flight path is assumed known in the form of azimuth and elevation points registered at each pulse. Arbitrary flight paths are permitted. The flight path is assumed to be such that the far-field plane wave propagation model is appropriate. The transmitter is assumed to transmit a waveform at center frequency F_c and bandwidth B_w. The waveform may be a chirp or any other waveform that demodulates to a phase history. The receiver is assumed to have a multiplicity $N_f > 1$ of frequency bins spaced equally from $F_c - B_w/2$ to $F_c + B_w/2$. The Jackson model supports multiple polarization, however we will only consider VV (vertical-vertical) polarization. The estimation technique we develop here is not limited to VV polarization.

3.1.2 Noise Model.

The received signal is assumed to be corrupted by Additive White Gaussian Noise (AWGN) of known variance σ^2. The noise samples are assumed to corrupt the I and Q channels independently. This received noise models all noise including thermal noise, receiver electronic noise, and atmospheric noise. Ground clutter, coherent interferers, colored noise, multi-path and jamming are not considered.

3.1.3 Target Model Assumptions.

The target scene is assumed to contain a known number of canonical shapes of unknown type and unknown parameters. The method of determining the number of shapes is not considered in this research but could potentially make use of imaging techniques described in [1, 24] . All shapes are assumed to be large enough to be within the resolution of the radar platform and small enough to be outside of aliasing zones. The received phase history is assumed to be the coherent sum of individual shape phase histories, and so the effects of shadowing and multiple-bounce effects (except those modeled by the canonical shapes themselves) are ignored.

We assume that, within a shape, the parameters themselves are statistically independent, identically distributed (IID). As a result we treat each parameter of a shape without regard for the values of the others.

Finally, for the purposes of this research, only mononstatic radars are explored, meaning that for all modeled shapes the transmitter and receiver locations are the same. Note that the methods developed bear no regard to this assumption and could be extended in the future to explore bistatic cases.

3.1.4 Integration with Automatic Target Recognition.

The ATR software is assumed to be a higher level system which provides the prior distributions of parameters. The estimator described in this research provides estimates of the type and parameters of each shape, as well as a measure of the 'confidence' of each estimate.

3.2 Bounds on the Parameter Space

The majority of this research involves the solution of probability integrals. In a naive form the domain of parameter space (length, width, radius, X, Y, Z, roll, pitch, yaw) is infinite. In practice, the finite radar bandwidth and discrete sampling of digital electronics presents aliasing, and certain parameters (such as the size parameters) are inherently positive. The bounds of the parameters become very important in the solution of the integral and must be carefully considered. Should parameter bounds be taken outside these limits, aliasing may occur in interval estimates, and incorrect constants may be calculated for, e.g. marginalization and the Bayesian denominator.

3.2.1 Position Parameters.

The canonical shapes described in [1], in general, are factored as a size- and pose-dependent magnitude term $M_{shape}(\underline{\mathbf{\Theta}}_{size}, \underline{\mathbf{\Theta}}_{pose})$ multiplied by a location-dependent phasor $\exp(-j\underline{\mathbf{k}} \bullet \underline{\mathbf{r}})$. The minimum position spacing is given by the range resolution as ([25], pg. 23)

$$\rho_u = \frac{c}{2B_w}, \tag{3.1}$$

and the minimum cross-range spacing is given as ([25], pg. 23)

$$\rho_y = \frac{4\pi}{\lambda} 2 \sin\left(\frac{\Delta\phi}{2}\right), \tag{3.2}$$

where $\Delta\phi$ is the azimuthal extent of the flight path and λ is the wavelength of the carrier frequency.

Jakowatz [25] shows that the target scene is of fixed extent D, is given by

$$D_{max} = \frac{\lambda_{min}}{2\Delta\phi}, \tag{3.3}$$

where $\Delta\phi$ is the distance between azimuthal flight path samples.

3.2.2 Size Parameters.

We require that the shapes be located within the target scene, therefore the size of the object must be such that it is smaller than the scene diameter. There are added complexities when the effects of pose angle are considered, since the effective length of the object (which causes aliasing) may be different than the actual scene extent D_{max}.

3.2.3 Pose Parameters.

As noted above, there is coupling between size and pose angles. However, more importantly, the Jackson models described in Chapter 2 often have limits on the azimuth and elevation angles ϕ_t, ϕ_r, θ_t and θ_r. The mapping of these angles onto the roll-pitch-yaw parameters is not straightforward. Further research may focus on methods for determining these bounds.

3.3 Sampling the Posterior Density

Bayes' Rule (Equation (2.21)) provides a method of calculating a posterior distribution $\boldsymbol{p}(\underline{\boldsymbol{\Theta}} \,|\, \underline{\boldsymbol{y}})$ (which is directly related to the parameters in question) from the priors $\boldsymbol{p}(\underline{\boldsymbol{\Theta}})$ (defined regardless of any observation) and the likelihood $\boldsymbol{p}(\underline{\boldsymbol{y}} \,|\, \underline{\boldsymbol{\Theta}})$ (which is known from the observed data). However, the posterior is a probability density function (PDF) and so gives only values for specific parameter sets $\underline{\boldsymbol{\Theta}}$. Additionally, the probability of any specific parameter set is zero. Therefore, the posterior density is infinitely dense, and has high dimensionality even for a small number of shapes.

Silverman [26] discusses methods of using a small number of samples to interpolate the density function. Our research focuses on the opposite: given a density function, find a small set of values that encompass the important features of that density. Ideally, it is desirable to have a 'multi-zoom' capability so that increasing the number of points shows finer detail of the density. Such a multi-zoom capability can be achieved through the use of probability mass intervals.

3.3.1 Probability Mass Intervals.

Recall the definition of a probability density function $\boldsymbol{p}(X)$. The probability $\boldsymbol{P}$ that the random variable X is between values a and b is given by ([4])

$$\boldsymbol{P}(a \leq X \leq b) = \int_a^b \boldsymbol{p}(X)dX. \tag{3.4}$$

Let $[a, b]$ to be the *interval* and the probability $\boldsymbol{P}(a \leq X \leq b)$ to be the *probability mass* on the interval.

Consider an interval $[a, b]$ where $\boldsymbol{p}(X)$ has a high value somewhere inside. This intuitively corresponds to a region of high probability. Further, let the regions outside the interval $[a-\epsilon_1, a]$ and $[b, b+\epsilon_2]$, $\epsilon_{1,2} \geq 0$ be areas where $\boldsymbol{p}(X)$ has low value. The density $\boldsymbol{p}(X)$ must be nonnegative since it is a true probability density. Therefore,

$$\begin{aligned}
\boldsymbol{P}(a-\epsilon_1 \leq X \leq b+\epsilon_2) &= \int_{a-\epsilon_1}^{b+\epsilon_2} \boldsymbol{p}(X)dX \\
&= \underbrace{\int_{a-\epsilon_1}^{a} \boldsymbol{p}(X)dX}_{\text{(small)}} + \underbrace{\int_{a}^{b} \boldsymbol{p}(X)dX}_{\text{(large)}} + \underbrace{\int_{b}^{b+\epsilon_2} \boldsymbol{p}(X)dX}_{\text{(small)}} \qquad (3.5) \\
&\geq \int_{a}^{b} \boldsymbol{p}(X)dX \\
&\geq \boldsymbol{P}(a \leq X \leq b). \qquad (3.6)
\end{aligned}$$

Equation (3.5) shows that increasing the size of the interval increases the probability mass. Similarly, should one of the areas $[a-\epsilon, a]$ or $[b, b+\epsilon]$ be a region where the probability density is high (corresponding to a second high-probability region) the new probability mass would again be larger.

3.3.2 Bayesian Cost and Maximization.

The estimation problems that this research focuses on are related to finding peaks of the posterior density $\boldsymbol{p}(\underline{\boldsymbol{\Theta}} \,|\, \underline{\boldsymbol{y}})$. Peak finding can also be viewed as finding intervals of large probability mass. By the results of Equation (3.5) a large probability mass over a certain interval means there is possibly a sub-interval inside that also contains a large probability mass.

We will now show that the peak finding is analogous to the standard maximum a posteriori (MAP) estimate described in ([5], Ch. 11). Consider the function $C_h(\xi)$ to be the ‘cost’ of an estimated parameter $\hat{\theta}$ that is some distance ξ away from the true parameter θ, $\xi = \theta - \hat{\theta}$. The average cost $\boldsymbol{E}[C_h(\xi)]$ is called the *Bayes risk*. An optimum Bayesian estimator is one that minimizes this risk.

Following [5], we consider the hit-or-miss cost function. Let $C_{hm}(\xi)$ be zero cost if the estimate is within a 'hit width' $\pm\epsilon$ from the true value, and unit cost outside:

$$C_{hm}(\xi) = \begin{cases} 0 & |\xi| \leq \epsilon \\ 1 & |\xi| > \epsilon. \end{cases} \tag{3.7}$$

The Bayes risk for this cost is then the expected value with respect to the joint probability $\boldsymbol{p}(\underline{\boldsymbol{y}}, \theta)$,

$$\begin{aligned} \boldsymbol{E}[C_{hm}(\xi)] &= \int_{\mathbb{R}} \int_{\mathbb{R}} C_{hm}(\theta - \hat{\theta}) \boldsymbol{p}(\underline{\boldsymbol{y}}, \theta) d\underline{\boldsymbol{y}} d\theta \\ &= \int_{\mathbb{R}} \int_{\mathbb{R}} C_{hm}(\theta - \hat{\theta}) \boldsymbol{p}(\theta \,|\, \underline{y}) \boldsymbol{p}(\underline{y}) d\underline{\boldsymbol{y}} d\theta \\ &= \int_{\mathbb{R}} \left[\int_{\mathbb{R}} C_{hm}(\theta - \hat{\theta}) \boldsymbol{p}(\theta \,|\, \underline{y}) d\theta \right] \boldsymbol{p}(\underline{y}) d\underline{\boldsymbol{y}}. \end{aligned} \tag{3.8}$$

The last step makes use of the Fubini Theorem ([14], pg. 61) to change the order of integration.

Following [5], we will minimize the Bayes risk by minimizing the inner integral of Equation (3.8). Substituting the definition of the cost function in Equation (3.7), the minimization of Equation (3.8) becomes

$$\begin{aligned} \arg\min_{\hat{\theta}} \boldsymbol{E}[C_{hm}(\xi)] &= \arg\min_{\hat{\theta}} \int_{\mathbb{R}} C_{hm}(\theta - \hat{\theta}) \boldsymbol{p}(\theta \,|\, \underline{y}) d\theta \\ &= \arg\min_{\hat{\theta}} \int_{-\infty}^{\hat{\theta}-\epsilon} \boldsymbol{p}(\theta \,|\, \underline{y}) d\theta + \int_{\hat{\theta}+\epsilon}^{\infty} \boldsymbol{p}(\theta \,|\, \underline{y}) d\theta \\ &= \arg\min_{\hat{\theta}} \int_{-\infty}^{\infty} \boldsymbol{p}(\theta \,|\, \underline{y}) d\theta - \int_{\hat{\theta}-\epsilon}^{\hat{\theta}+\epsilon} \boldsymbol{p}(\theta \,|\, \underline{y}) d\theta \\ &= \arg\min_{\hat{\theta}} 1 - \int_{\hat{\theta}-\epsilon}^{\hat{\theta}+\epsilon} \boldsymbol{p}(\theta \,|\, \underline{y}) d\theta \\ &= \arg\max_{\hat{\theta}} \int_{\hat{\theta}-\epsilon}^{\hat{\theta}+\epsilon} \boldsymbol{p}(\theta \,|\, \underline{y}) d\theta. \end{aligned} \tag{3.9}$$

As seen in Equation (3.9), the optimum estimate $\hat{\theta}$ is a region where the posterior density $\boldsymbol{p}(\theta \,|\, \underline{y})$ is largest. In the typical treatment such as [5], the MAP estimate $\hat{\theta}_{MAP}$ is the limit as the 'hit' width $\epsilon \to 0$, which results in finding the *mode* of the posterior density.

However, if ϵ remains finite, the integral in Equation (3.9) bears striking resemblance to the definition of the probability mass in Equation (3.4) when a and b are replaced by $\theta \pm \epsilon$ and therefore provides a new meaning of the MAP estimate.

3.3.3 MAP Estimate for Probability Mass.

Note the analogy of Equation (3.9) to the probability mass of Equation (3.4) would assume the estimate $\hat{\theta}$ could ‘slide’ around a continuum of parameter estimates. We wish to develop an optimum estimator that is tailored to the discrete nature of probability mass intervals of Equation (3.5). Consider the intervals themselves and how they partition the posterior probability space. Define a new distance metric ξ_i' that is the distance from the ‘true’ parameter θ to a series of discrete, equally spaced points $\hat{\theta}_i$ in parameter space:

$$\xi_i' = \theta - \hat{\theta}_i \tag{3.10}$$

$$= \theta - 2i\epsilon. \tag{3.11}$$

The parameter ϵ is the interval half-width (specified by the user) and the index i is an integer that indexes which region of the parameter space the distance is being measured. For example, assume we have a one-dimensional parameter space to estimate X location (in meters) of a certain canonical shape, and all other parameters are ‘given’. At an interval of 0.1 m, the half-width ξ is 0.05 m. The values of ξ_i' are $\{...\theta - 0.15, \theta - 0.05, \theta + 0.05, \theta + 0.15, ...\}$ meters. The set ξ_i' form a sequence of discrete values for a given theta. As defined in Equation (3.10), the metric is an abstraction whose value only becomes concrete when a specific value of θ is given.

The result given in Equation (3.9) was very close to the probability mass definition, so we continue to use the hit-or-miss cost function defined in Equation (3.7) with the new distance metric ξ_i' . The optimum estimate is now a function of the index i since the spacing ϵ is assumed fixed. The resulting optimization problem is similar to the MAP estimate:

$$
\begin{aligned}
\arg\min_{\hat{\theta}_i} \boldsymbol{E}[C_{hm}(\xi_i')] &= \arg\min_{i} \boldsymbol{E}[C_{hm}(\xi_i')] \\
&= \arg\min_{i} \int_{\mathbb{R}} C_{hm}(\theta - 2i\epsilon)\boldsymbol{p}(\theta\,|\,\underline{y})d\theta \\
&= \arg\min_{i} \int_{-\infty}^{2i\epsilon-\epsilon} \boldsymbol{p}(\theta\,|\,\underline{y})d\theta + \int_{2i\epsilon+\epsilon}^{\infty} \boldsymbol{p}(\theta\,|\,\underline{y})d\theta \\
&= \arg\min_{i} \int_{-\infty}^{\infty} \boldsymbol{p}(\theta\,|\,\underline{y})d\theta - \int_{(2i-1)\epsilon}^{(2i+1)\epsilon} \boldsymbol{p}(\theta\,|\,\underline{y})d\theta \\
&= \arg\min_{i} 1 - \int_{(2i-1)\epsilon}^{(2i+1)\epsilon} \boldsymbol{p}(\theta\,|\,\underline{y})d\theta \\
&= \arg\max_{i} \int_{(2i-1)\epsilon}^{(2i+1)\epsilon} \boldsymbol{p}(\theta\,|\,\underline{y})d\theta \\
&= \arg\max_{i} \boldsymbol{P}(\ (2i-1)\epsilon \le \theta \le (2i+1)\epsilon\,|\,\underline{y}\ ). \qquad (3.12)
\end{aligned}
$$

Notice the last step is simply a probability mass calculation over a certain interval. Assume the user has already repeated the calculation of probability mass over each discrete interval in the parameter space, and has stored the corresponding mass in a set $\{P_i\}$.

$$P_i = \boldsymbol{P}(\ (2i-1)\epsilon \le \theta \le (2i+1)\epsilon\,|\,\underline{y}\ ). \qquad (3.13)$$

Then the result of Equation (3.12) shows that the optimum parameter range is

$$\arg\min_{\hat{\theta}_i} \boldsymbol{E}[C_{hm}(\xi_i')] = \arg\max_{i} P_i. \qquad (3.14)$$

Equation (3.14), and the corresponding derivation of Equation (3.12) that lead to it, are a very important result to this research. They provide a statement of the formal optimization problem and an intuitive result.

To summarize the results of this section, recall the steps leading to Equation (3.14). We have modified the hit-or-miss cost function $C_{hm}(\xi)$ to include a zero-cost region of width 2ϵ, and therefore *any* parameter estimate $\hat{\theta}$ within $\pm\epsilon$ of the true parameter θ is considered equally, perfectly, correct. It implies that no ‘better’ estimate of θ can be made than any other estimate within this region. We have also modified the distance metric ξ

to measure the distance between the true parameter θ and a set of discrete parameters. This new distance measure ξ_i' is equivalent to sampling parameter space and measuring the distance to the i^{th} sample parameter. The combination of the zero-cost region and discretized distance means that the optimization given in Equation (3.12) no longer seeks to find a best *parameter*, but instead seeks a *range* of parameters to which the true parameter is *closest*. This formalizes the notion that the optimum parameter range for a given input $\underline{\boldsymbol{y}}$ is the largest probability mass over a series of masses at equally-spaced intervals.

3.3.4 Probability Mass Interval Algorithm.

Beginning with a few disjoint large intervals. Each interval that has a large probability mass is subdivided into sub-intervals and their respective probability masses are calculated. Sub-intervals of low mass are discarded from further study. The process is repeated until the smallest sub-interval containing nearly all the mass is found. In this case ‘smallest’ means further sub-division either a) significantly reduces the mass, or b) creates sub-intervals that are adjacent and also have significant mass. Alternatively the algorithm terminates when a pre- defined ‘smallest interval’ has been reached. This is summarized below.

1. Set the initial interval to the bounds of the parameter (scene extent, largest shape, etc).
2. Partition the interval into *M* sub-intervals.
3. For each interval, calculate the probability mass.
4. If no interval contains any mass, subdivide into smaller intervals and goto step 3 otherwise goto step 5.
5. If already at the user-specified minimum interval, terminate.
6. Find each peak that meets some acceptance criteria. For each peak, subdivide that interval into smaller sub-intervals.
7. For each sub-interval, calculate the probability mass.
8. For each sub-interval, if there is no mass remove the sub-interval from further consideration.
9. Goto step 5.

3.3.5 Similarity with the Bayesian Denominator Calculation.

The same algorithmic steps used to create the Bayesian denominator in Equation (2.53) are also used in the calculation of the probability masses. Using the affine transformation defined in Equation (2.58), a new set of bounds is created for the Gauss-Legendre quadrature Equation (2.59). Further, since a change of bounds only requires a change in the abscissa points (not the weight coefficients or the steps to implement the quadrature), there is the potential for significant code re-use.

3.4 Calculation of Credible Region

The Gill definition in Equation (2.60) shows how statements can be made about the probability of the true parameters $\underline{\mathbf{\Theta}}$ being within a region in parameter space. A region C that has a high probability of containing $\underline{\mathbf{\Theta}}$ may be called a *credible region*. The credible region C is an abstract, D-dimensional region in parameter space, not to be confused with the Bayesian cost function $C_{hm}(\xi)$ of Section 3.3.2. There may be many regions of the parameter space which satisfy Equation (2.60). In this research, we consider parameter spaces of D dimensions corresponding to the D shape parameters (X,Y,Z, roll pitch, yaw, height, length, radius, etc.). The Gill definition only provides a single constraint: the total probability α of $\underline{\mathbf{\Theta}}$ being within the region, and this alone is insufficient to define C.

We consider two separate definitions of the region C that align with common notions of credibility (or confidence) based on if the parameter dimensions of $\underline{\mathbf{\Theta}}$ should be treated separately or not. We designate these two definitions as the *Credible Interval* and the *Credible Set*. In each case, we begin with a set of probability masses $P_{k_1,...k_D}$ that are the multidimensional analog of those calculated by Equation (3.13). Equation (2.60) requires that the combined probability mass over C be α.

3.4.1 Credible Set.

We postulate that any useful definition of a credible region must contain the highest-probability intervals of parameter space. Therefore we define the *credible set* as the indices of the M largest probability masses $P_{k_1,...k_D}$ such that their combined mass is α. Let $\boldsymbol{O}[P]$ denote the ordering of the probability masses from largest to smallest such that $\boldsymbol{O}[P]$ returns the indices $(k_1, ..., k_D)$ for the i^{th} largest mass

$$\boldsymbol{O}[P](i) : \mathbb{N} \mapsto \mathbb{N}^D \quad \text{such that} \quad i > j \implies P_{\boldsymbol{O}[P](i)} \geq P_{\boldsymbol{O}[P](j)}. \tag{3.15}$$

Then the credible set C_P can be defined based on the ordered masses as

$$C_P = \bigcup_{i=1}^{M} \boldsymbol{O}[P](i) \quad \text{such that} \quad \sum_{i=1}^{M} \boldsymbol{O}[P](i) = \alpha. \tag{3.16}$$

In practice, we cannot expect that the probability masses sum exactly to the prescribed α. The problem may be handled by defining *upper* and *lower* credible sets C_P^+ and C_P^-, respectively. The sets are defined as

$$C_P^+ = \bigcup_{i=1}^{M} \boldsymbol{O}[P](i) \quad \text{such that} \quad \sum_{i=1}^{M-1} \boldsymbol{O}[P](i) < \alpha \leq \sum_{i=1}^{M} \boldsymbol{O}[P](i), \tag{3.17}$$

and

$$C_P^- = \bigcup_{i=1}^{M} \boldsymbol{O}[P](i) \quad \text{such that} \quad \sum_{i=1}^{M} \boldsymbol{O}[P](i) \leq \alpha < \sum_{i=1}^{M+1} \boldsymbol{O}[P](i). \tag{3.18}$$

The upper credible set C_P^+ has at most one 'extra' probability mass, making it slightly too big. Likewise the lower credible set C_P^- has at most one 'missing' mass, making it slightly too small. Thus C_P^+ and C_P^- bound the region and (combined with the multizoom techniques described in Chapter 4) could potentially be useful in certain interval analysis techniques such as SIVIA described in [27].

3.4.2 Credible Intervals.

The ideal credible set C_P defined in Equation (3.16) is likely to be a region of arbitrary shape, and fundamentally is a statement about the entire parameter space. Because the parameters considered in this research are presumed independent (with the exception noted in Section 4.3), it seems reasonable that we make statements about individual parameter dimensions separately. We call this the *Credible Interval*. For example, the user may wish to know about the credible region when considering only the intervals of the X location parameter, located at the j^{th} ordinate among the probability masses. In this case the credibility is taken over the marginal probability mass set $\left\{P_{k_j}\right\}$. Because probability masses are true probabilities (the denominator in Bayes' Rule has been calculated) the marginalization is simply a sum along the other dimensions. Consider the case where we wish to know the credible interval along dimension j. The marginal probability mass set

$\left\{P_{k_j}\right\}$ is

$$P_{k_j} = \sum_{k_1} \cdots \sum_{k_{j-1}} \sum_{k_{j+1}} P_{k_1 \dots k_D} \tag{3.19}$$

The marginal mass set $\left\{P_{k_j}\right\}$ is just a new set of probability masses, with all the features discussed above. Therefore, the credible interval is, in fact, the credible region along a marginal mass set. We denote this interval C_{P_j}.

3.5 GPU Implementation

For increased speed, several of the core functions are implemented in the CUDA GPU. The Gauss-Legendre quadrature Equation (2.59) requires repeated evaluation of the Jackson model phase history functions described in Section 2.1. These functions perform largely the same calculation for a given parameter set $\underline{\boldsymbol{\Theta}}$: repeately apply the canonical shape magnitude M and phase calculations for all azimuth and elevation angles θ, ϕ, over a range of wave numbers k.

The CUDA implementation of the file `Model_PhaseHistory` from [1] constructs a 3-dimensional array: a dimension corresponding to azimuth/elevation points, a dimension corresponding to wavenumber k, and a dimension corresponding to the parameter set for a given abscissa in Equation (2.59), as shown in Figure 3.1. The slices of k and azimuth/elevation represent 'test phase histories' to be applied to the likelihood function next. The CUDA implementation uses a separate PTX file for each canonical shape to overcome MATLAB's inability to resolve multiple *de-mangled* C++ function names. C++ allows the programmer to code several implementations of a function that all have the same name (for example the function `add()` may be defined for scalar or vector data types. This process is known as *overloading* [28]. The C++ compiler re-names the functions internally in a form known as the *mangled* name [29]. Our experience with Matlab 2012b is that it has trouble with these names. This results in errors trying to use a single PTX file for all phase history models.

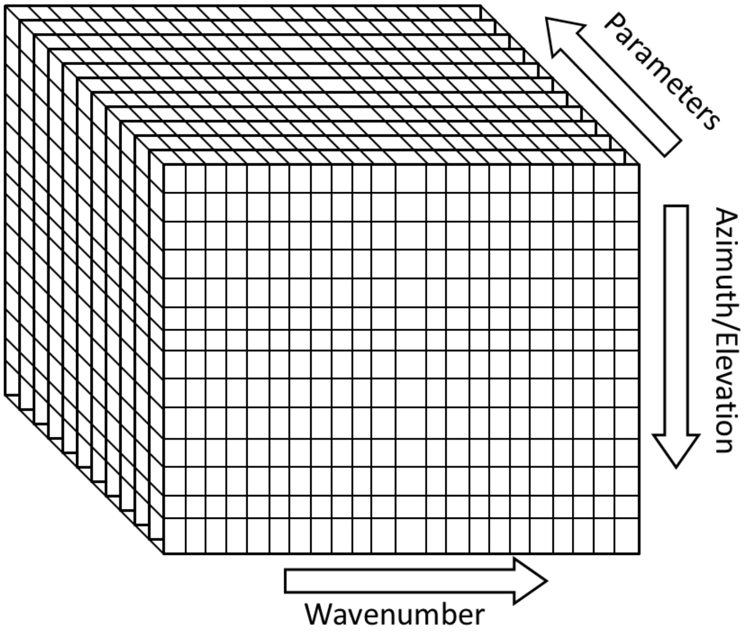


Figure 3.1: Layout of phase history data blocks.

A second CUDA function performs the combined subtraction and inner product in Equation (2.67). The test phase histories are always subtracted from the same received data vector $\underline{\boldsymbol{y}}$. The vector $\underline{\boldsymbol{y}}$ is stored in the GPU high-speed shared memory.

A final CUDA function applies the weight coefficients $c_{k_1} ... c_{k_D}$ and the prior distribution $\boldsymbol{p}(\underline{\boldsymbol{\Theta}})$. The result is returned to the main memory for use in displays and the credible interval calculation.

IV. Results and Analysis

4.1 Validation of GPU Code

This section describes the results of validating the implementation of the Jackson `Model_PhaseHistory` MATLAB file in the CUDA graphics processing unit (GPU). As mentioned Section 3.5, a separate CUDA code file is created for each of the canonical shapes. To ensure consistency with the Jackson implementation (which is considered to be the baseline) each PTX file is subjected to a verification process. In this process a series of 10,000 trials, each with a random realization of position (X, Y, Z), size (radius, length, height), and pose (roll, pitch, yaw) for a given shape. Only those parameters appropriate for a shape are applied. The simulation of, for example, the radius for a plate shape is not performed. The calculations for radar cross-section (RCS) are given in the tables of Section 4.2. These parameter realizations are applied to both the Jackson `Model_Phasehistory` file and the CUDA implementation.

The resulting phase histories for each parameter set are compared sample-by-sample, and two quality metrics are extracted for the entire realization: maximum absolute difference over the phase history, and mean absolute difference over the phase history. The sample-by-sample difference of the Jackson and CUDA implementations are normalized by RCS so that different parameter realizations can be compared without regard to scale factor. Figures 4.1-4.6 show the results of the simulations. The figures may be interpreted as both an overall error between the CUDA and Jackson implementations (by examining the trend over all trials) and as an indicator of potentially problematic combinations of parameters (by observing high error isolated to a single trial). Figures 4.1-4.6 show that the (normalized) mean error is on the order of 10^{-5}. This is considered sufficient for the purposes of this research since the noise applied in the detection scenarios is well above 10^{-5}. Future research may focus on the examining the differences between MATLAB and

CUDA in the numerical implementation of key transcendental functions such as `sin(x)`, `cos(x)` and `atan2(y,x)`.

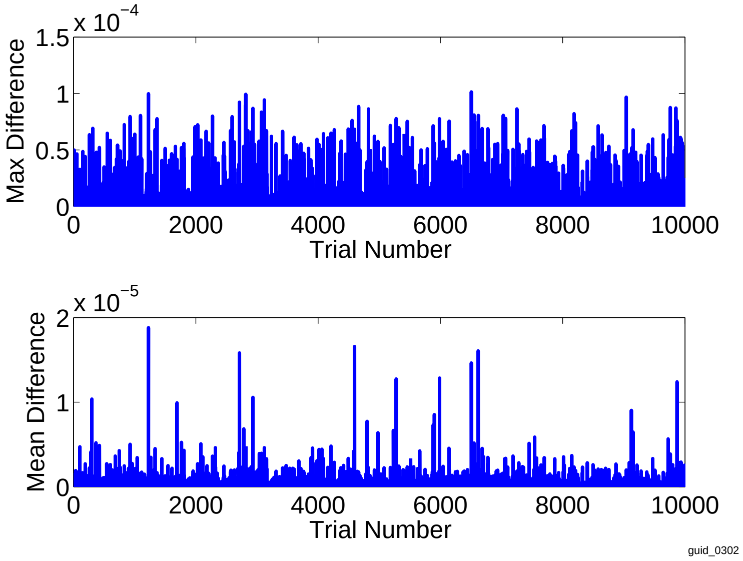


Figure 4.1: CUDA implementation error for plate (Top) measured using max absolute difference, and (Bottom) measured using mean absolute difference.

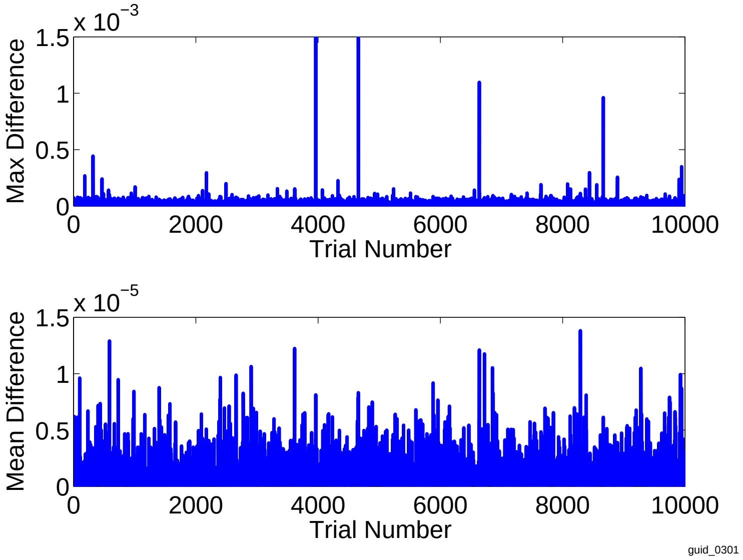


Figure 4.2: CUDA implementation error for dihedral (Top) measured using max absolute difference, and (Bottom) measured using mean absolute difference.

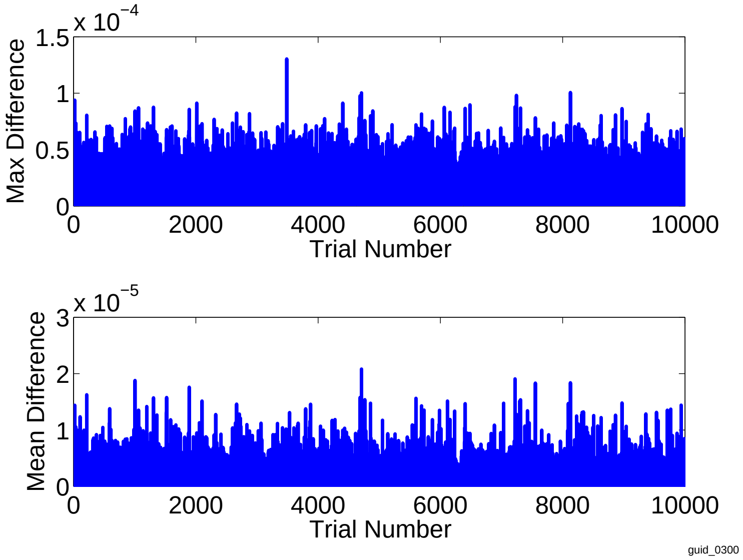


Figure 4.3: CUDA implementation error for top-hat (Top) measured using max absolute difference, and (Bottom) measured using mean absolute difference.

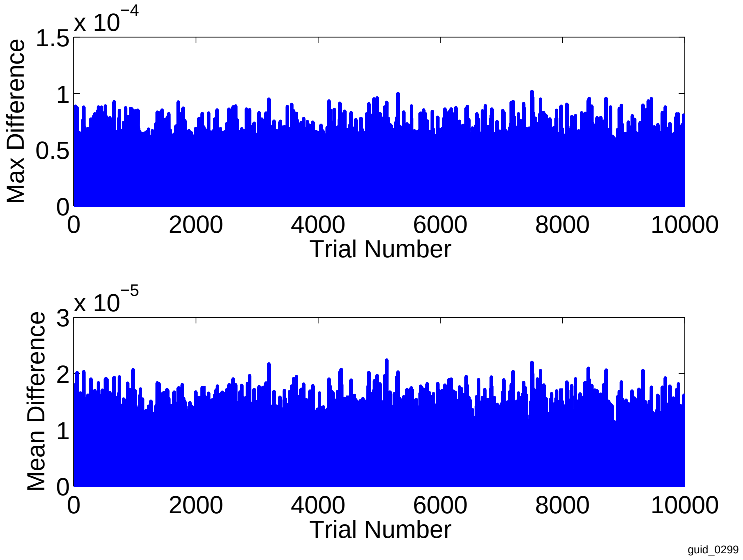


Figure 4.4: CUDA implementation error for sphere (Top) measured using max absolute difference, and (Bottom) measured using mean absolute difference.

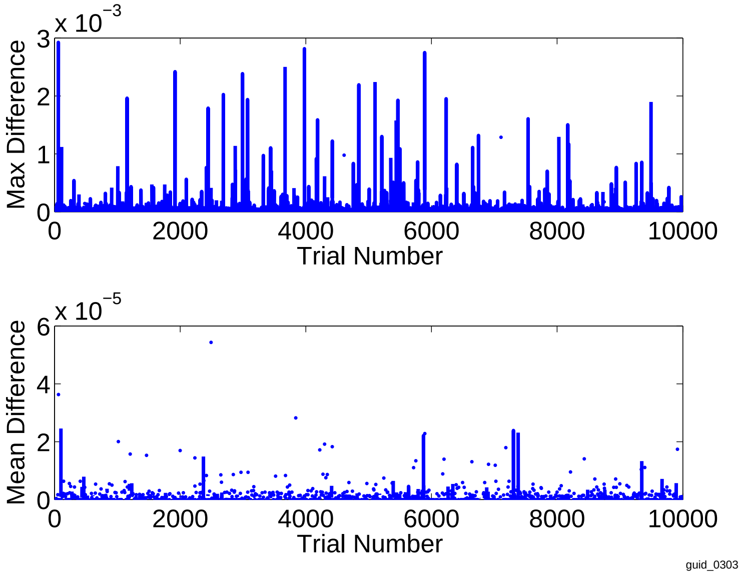


Figure 4.5: CUDA implementation error for cylinder (Top) measured using max absolute difference, and (Bottom) measured using mean absolute difference.

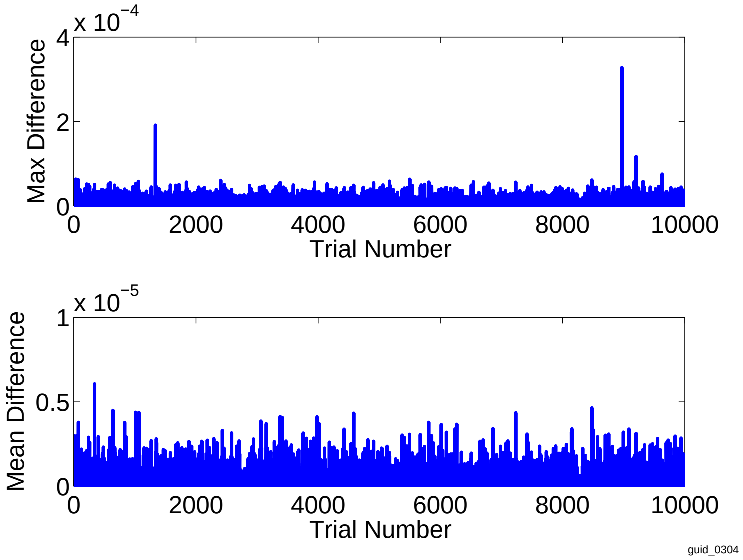


Figure 4.6: CUDA implementation error for trihedral (Top) measured using max absolute difference, and (Bottom) measured using mean absolute difference.

4.2 Estimation with Uniformative Priors

This section describes the results of parameter estimation with uniformative priors, i.e. $\boldsymbol{p}(\underline{\boldsymbol{\Theta}})$ is some constant such that $\int_{\mathbb{R}} \boldsymbol{p}(\underline{\boldsymbol{\Theta}})d\underline{\boldsymbol{\Theta}} = 1$. For several scenarios, a test shape is generated with additive noise. Unless otherwise specified, the simulation parameters used are shown in Table 4.1.

Table 4.1: Simulation Parameters for Uninformative Priors.

Parameter	**Value**	**Units**
Azimuth	-90:1:90	degrees
Elevation	30	degrees
Carrier Frequency	300	MHz
Bandwidth	100	MHz
Frequency Bins	64	Bins
Noise Variance	0.01	
Polynomial Order	5	

4.2.1 Single Plate Shape.

The interval estimation is performed over a series of known plate shapes. The plate shape has eight parameters: (X,Y,Z), (roll, pitch, yaw) and (height H, length L). The RCS area is calculated as $A = HL$. At each simulation, two shape parameters are estimated and the remaining are 'given' a-priori. This is equivalent to perfect knowledge of the other 'nuisance' parameters. Unless otherwise specified, the simulation parameters used for the 'nuisance' parameters are shown in Table 4.2.

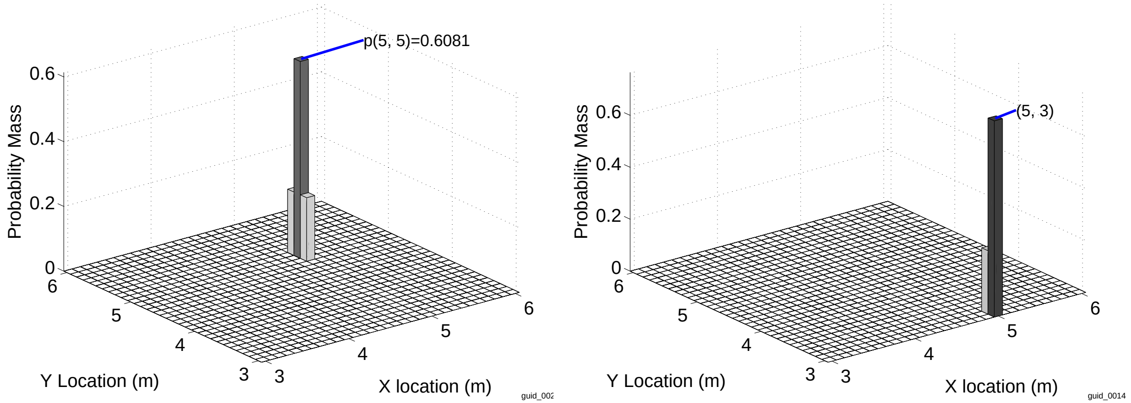
4.2.1.1 Plate Position Estimation.

Figure 4.7 shows the result of estimating the position of two different plates. In Figure 4.7(a), the plate's true position is located at $X = 5, Y = 5$. The estimation of (X, Y) shows a sharp spike at $(5, 5)$ when performed over a 0.1 m interval. Nearly all the probability mass is contained at the correct parameter values. Figure 4.7(b) is the same simulation with the true position moved to $(X, Y) = (5, 3)$. The estimator finds most of

Table 4.2: Single Plate Nuisance Parameters.

Name	Value	Unit
X	5	meters
Y	6	meters
Z	0	meters
Length	2	meters
Height	6	meters
RCS	$L \times H$	sq. meters
Roll	5	degrees
Pitch	20	degrees
Yaw	0	degrees
Polarization	VV	

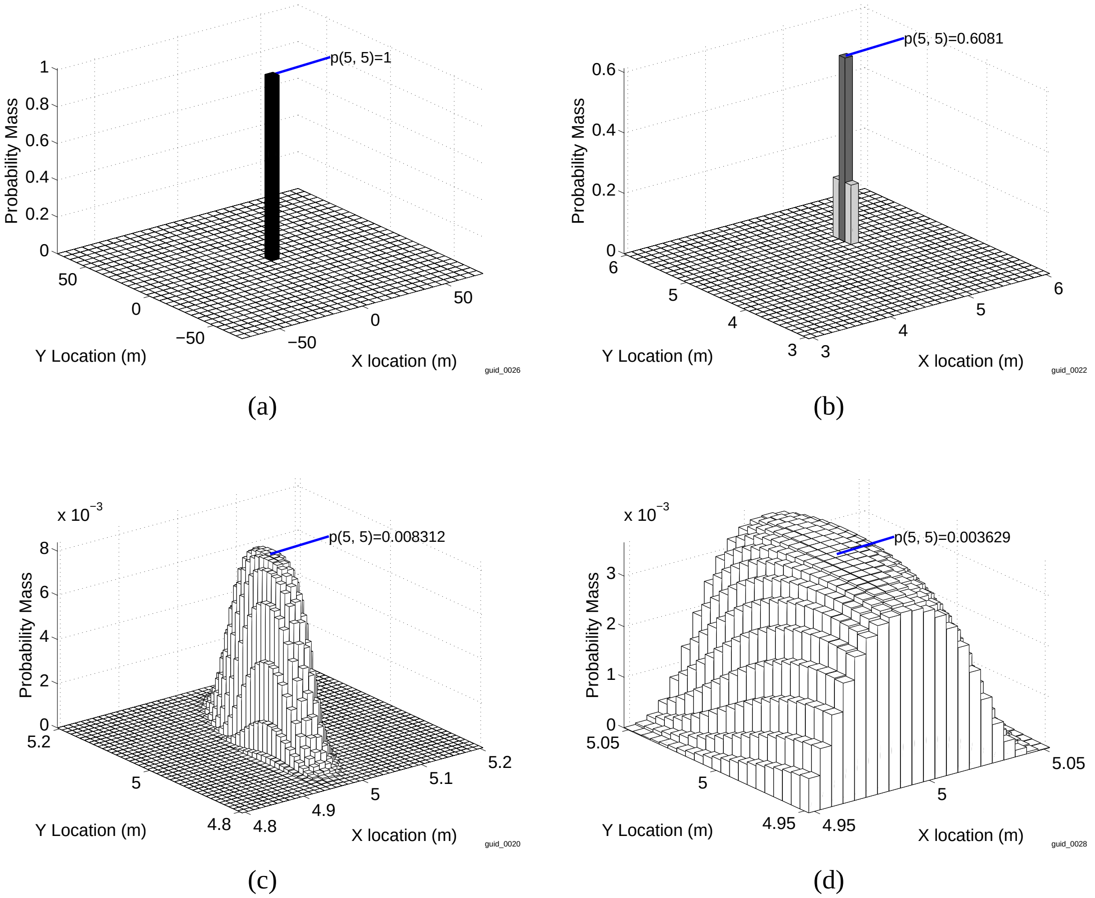
the probability mass at the new parameter values. Figure 4.7 is used as a ‘sanity check’ to ensure the interval estimator is working properly.



(a) (b)

Figure 4.7: Validating estimation of the X and Y position a single plate shape located at position (a) [5,5], and (b) [5,3], over a 0.1 m interval.

Figure 4.8 shows a series of progressively finer interval estimates to demonstrate the multi-zoom capabilities of the interval estimator. At first, a very wide area of the (X, Y) parameter space (100 m) is evaluated using a coarse interval (5 m). A large probability mass is located near $(X, Y) = (5, 5)$. The interval is decreased, as are the bounds of parameter space. In each step only the area of parameter space around the largest peak becomes the new bounds for the next finer interval. As the interval is increased, the fine structure of parameter space is revealed.



(a)

(b)

(c)

(d)

Figure 4.8: Estimation of the X and Y position a single plate shape located at position $X = 5, Y = 5$ at intervals of (a) 5 m, (b) 1 m, (c) 0.1 m, and (d) 0.005m.

4.2.1.2 Plate Size Estimation.

This section tests the results of the estimator to determine size parameters (length and height). Figure 4.9 shows the results of progressively finer interval estimates. Again, at first a coarse interval (1 m) and a large parameter space (L, H) of 50 m is used at first. The probability mass is concentrated at one point. The bounds are centered around this point in parameter space with a finer interval. The estimator consistently picks out the parameter. Again, the fine structure is evident at the increased zoom.

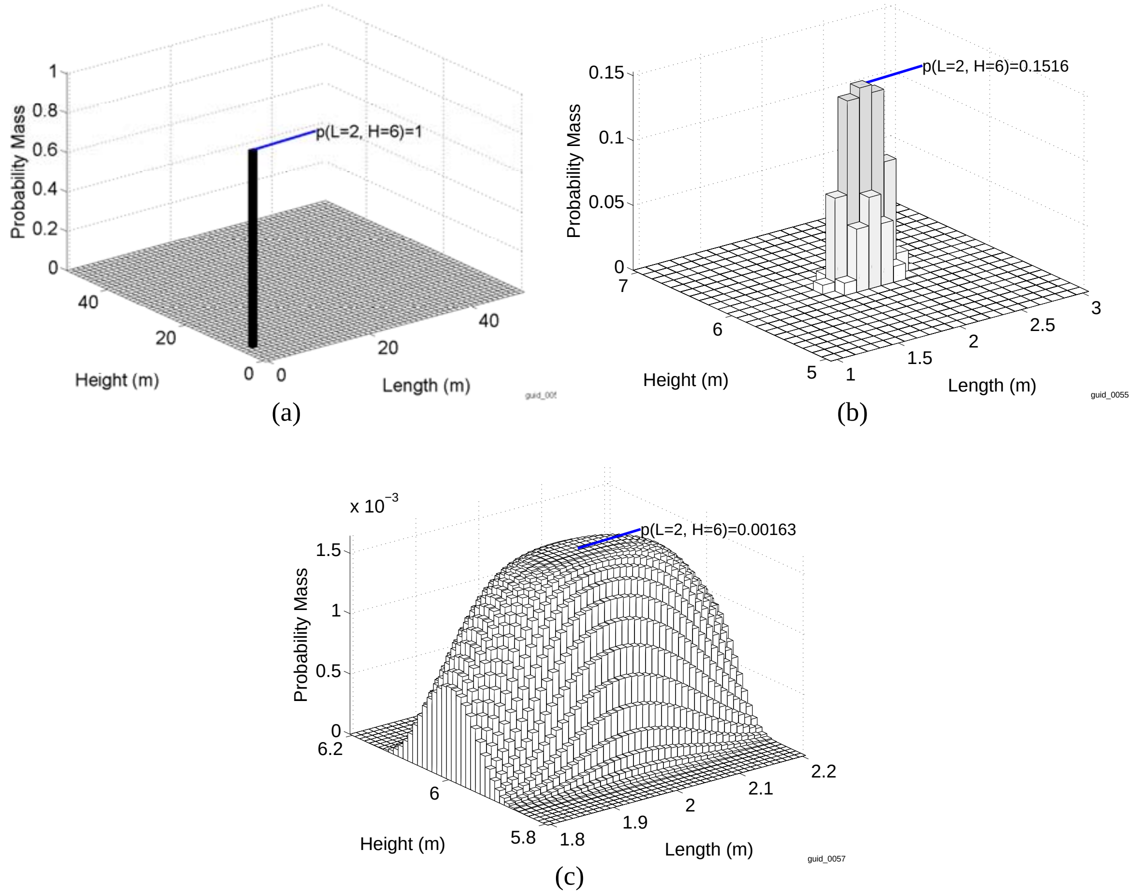
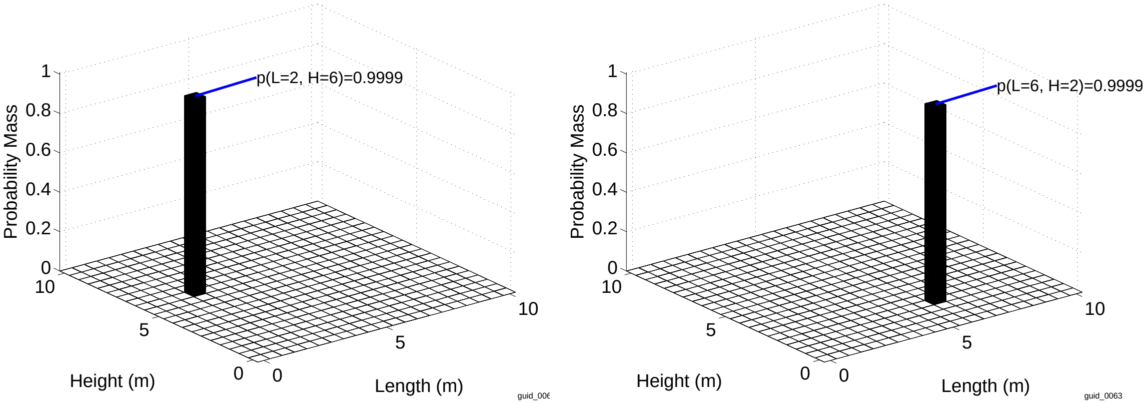


Figure 4.9: Estimation of the length and height parameters of a single plate shape of true size $L = 4, H = 2$ at intervals of (a) 1m, (b)0.1m, and (c) 0.01m.

The labels of the parameters for 'length' and 'height' are somewhat arbitrary. The definition of each is dependent on the exact pose in question. One estimate of 'height' is another estimate's 'length'. If the pose is perfectly known (as it is in these 2-parameter simulations) the labeling returned by the estimator matches that of the parameter set used to generate the received signal. However, if the pose is given erroneously, the resulting length and height may be swapped as shown in Figure 4.10. In this simulation two given values of roll are used: one with the true roll, and another where the test roll is 90° offset. These simulations have all non-estimated parameters as 'given'. Note that this simple (albeit severe) error moves the estimated length and height in parameter space to their corresponding opposites from $L = 2, H = 6$ to $L = 6, H = 2$.

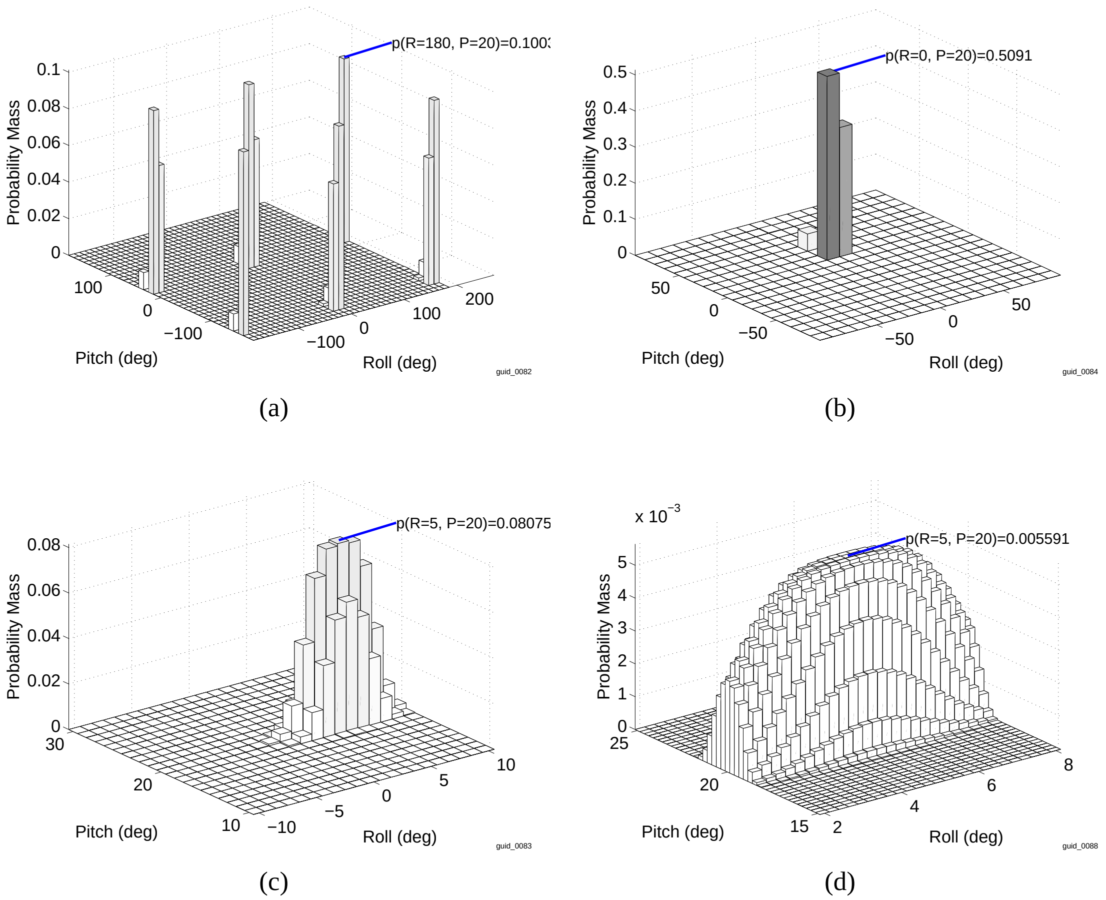


(a) (b)

Figure 4.10: Pose effects on length and height parameters of single plate of size $L = 2$, $H = 6$ at (a) True pose, and (b) 90° roll offset from True.

4.2.1.3 Plate Pose Estimation.

Figure 4.11 shows the results of successively finer intervals on estimation of the pose parameters roll and pitch (estimating yaw is similar). Note that in Figure 4.11(a) the bounds of parameter space are taken to be $-180°...180°$ as this is the most intuitive case. However it is clear from the plot of the interval masses that there is ambiguity in the parameters. This is due to the 180° ambiguity of the plate shape. A plate looks identical from the back than the front. Therefore the domain of pose parameters is restricted to $-90°...90°$. Again the fine structure is present at increased zoom.



(a)

(b)

(c)

(d)

Figure 4.11: Estimation of roll and pitch parameters for a plate at roll=5, pitch=20 at intervals of (a) 10°, (b) 10°, (c) 1°, and (d) $\frac{1}{4}^{\circ}$.

4.2.2 Single Sphere Shape.

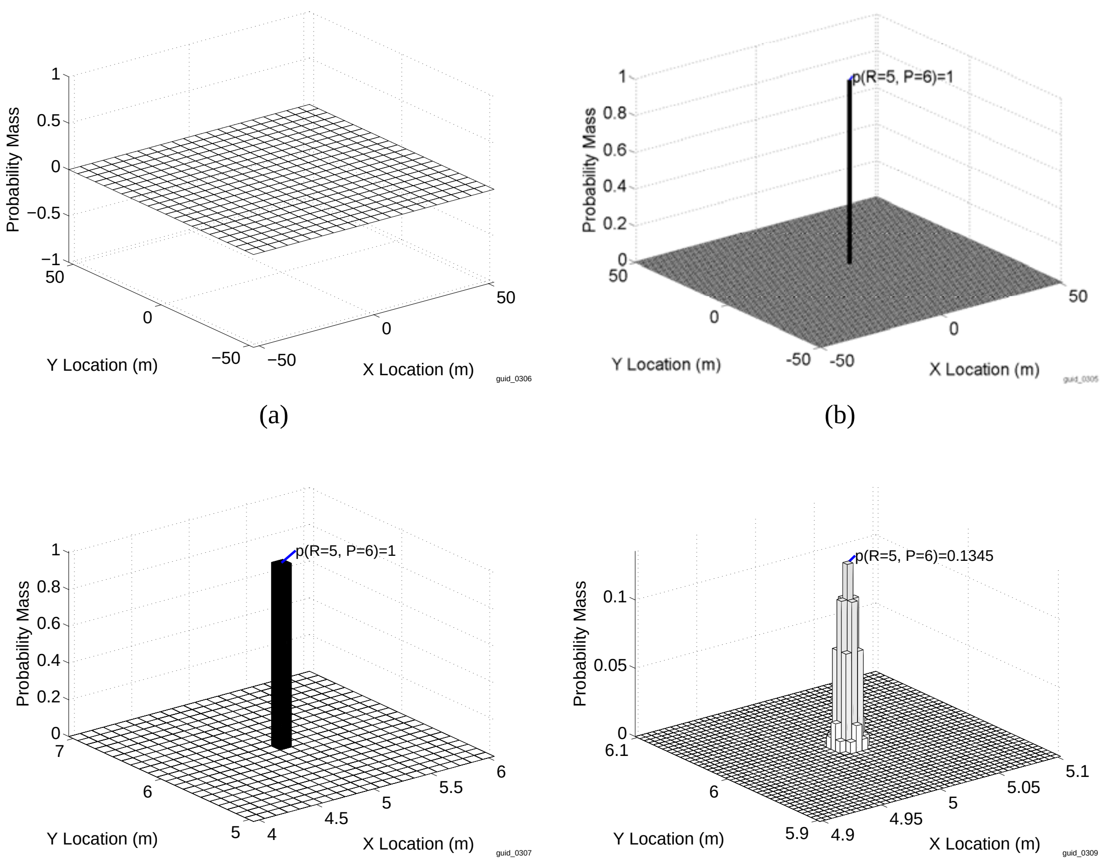
The interval estimation is performed over a series of known sphere shapes. The sphere shape has four parameters: (X,Y,Z) and (radius R). The RCS area is calculated as $A = R$. At each simulation, two shape parameters are estimated and the remaining are 'given' a-priori. This is equivalent to perfect knowledge of the other 'nuisance' parameters. Unless otherwise specified, the simulation parameters used for the 'nuisance' parameters are shown in Table 4.3.

Table 4.3: Single Sphere Nuisance Parameters.

Name	**Value**	**Unit**
X	5	meters
Y	6	meters
Z	0	meters
Radius	1.5	meters
RCS	R	sq. meters
Polarization	VV	

4.2.2.1 Sphere Position Estimation.

Figure 4.12 shows a series of progressively finer interval estimates to demonstrate the multi-zoom capabilities of the interval estimator. At first, a very wide area of the (X, Y) parameter space (100 m) is evaluated using a coarse interval (5 m) however the numerical precision is insufficient and no probability masses are found in the entire parameter space. The interval is decreased to 1 m with the same 100 m parameter space and a large probability mass is located near $(X, Y) = (5, 6)$. The interval is decreased, as are the bounds of parameter space. In each step only the area of parameter space around the largest peak becomes the new bounds for the next finer interval. As the interval is increased, the fine structure of parameter space is revealed.



(a)

(b)

(c)

(d)

Figure 4.12: Estimation of the X and Y position of a single sphere shape located at position $X = 5, Y = 6$ at intervals of size (a) 5 m, (b) 1 m, (c) 0.1 m, and (d) 0.005m.

4.2.2.2 Sphere Size Estimation.

This section tests the results of the estimator to determine size parameter (radius). Figure 4.13 shows the results of progressively finer interval estimates. Again, a coarse interval (0.5 m) and a large parameter space R of (50 m) is used at first. The probability mass is concentrated at one point. The bounds are centered around this point in parameter space with a finer interval. A minimum interval of 0.5 m is required before probability mass is detected; wider intervals do not detect any parameters while smaller intervals consistently pick out the parameter. Again, the fine structure is evident at the increased zoom.

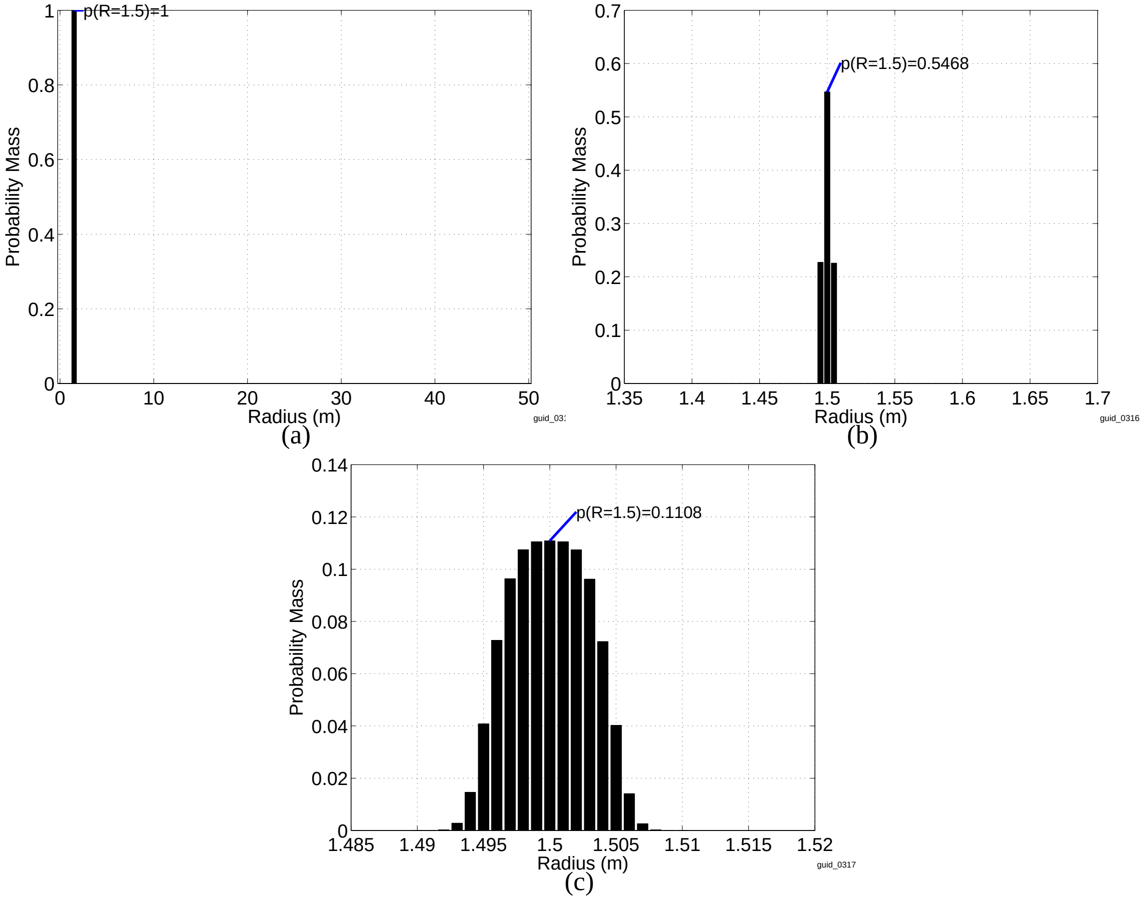


Figure 4.13: Estimation of the radius parameter of a single sphere shape of true size $R = 1.5$ at intervals of (a) 0.5m, (b)0.05m, and (c) 0.001m.

4.2.2.3 Sphere Pose Estimation.

The sphere shape phase history has no pose parameters because the phase history is spherically symmetric.

4.2.3 Single Dihedral Shape.

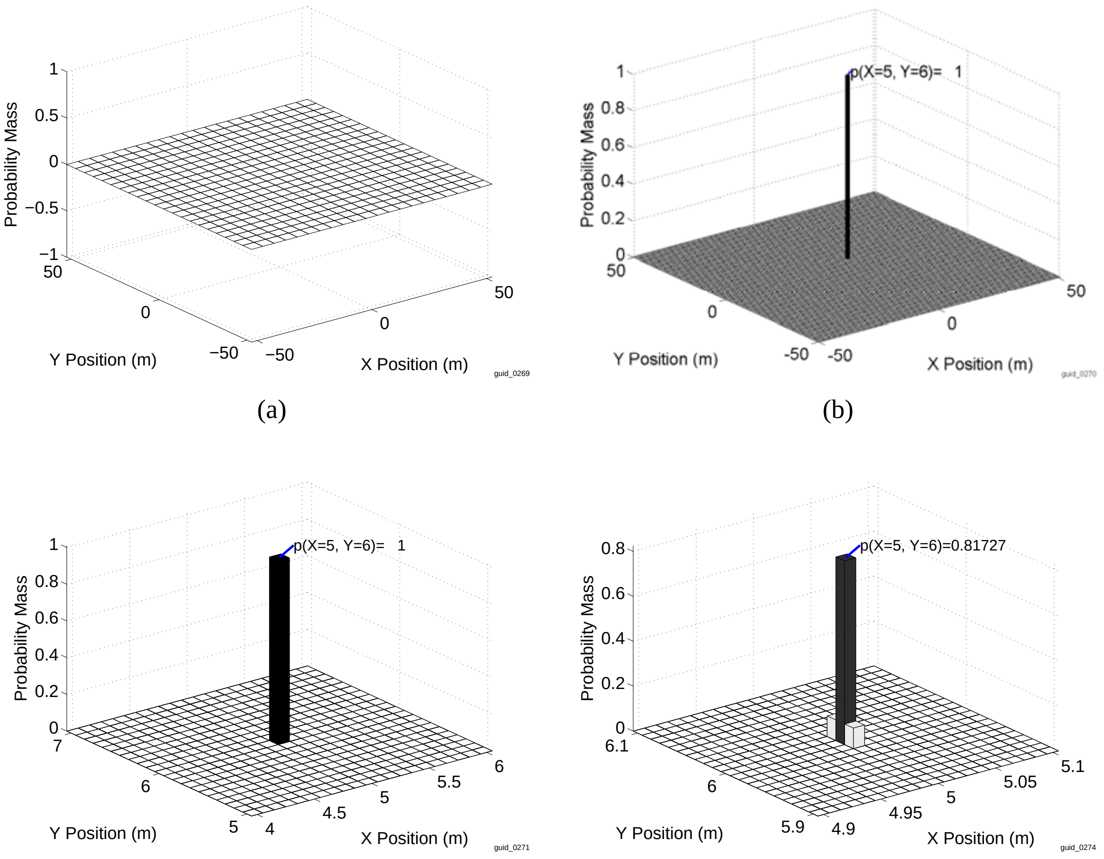
The interval estimation is performed over a series of known dihedral shapes. The dihedral shape has eight parameters: (X,Y,Z), (roll, pitch, yaw) and (height, length). The RCS area is calculated as $A = 2HL$. At each simulation, two shape parameters are estimated and the remaining are 'given' a-priori. This is equivalent to perfect knowledge of the other ‘nuisance’ parameters. Unless otherwise specified, the simulation parameters used for the ‘nuisance’ parameters are shown in Table 4.4.

Table 4.4: Single Dihedral Nuisance Parameters.

Name	Value	Unit
X	5	meters
Y	6	meters
Z	0	meters
Length	2	meters
Height	6	meters
RCS	$2 \times L \times H$	sq. meters
Roll	0	degrees
Pitch	30	degrees
Yaw	0	degrees
Polarization	VV	

4.2.3.1 Dihedral Position Estimation.

Figure 4.14 shows a series of progressively finer interval estimates of the position space. At first, a very wide area of the (X, Y) parameter space (100 m) is evaluated using a coarse interval (5 m). This resolution is insufficient since the estimator finds no masses anywhere, so the resolution is increased to 1 m. A large probability mass is located near $(X, Y) = (5, 6)$. The interval is decreased as are the bounds of parameter space. In each step, only the area of parameter space around the largest peak becomes the new bounds for the next finer interval. As the interval is increased, the fine structure of parameter space is revealed.



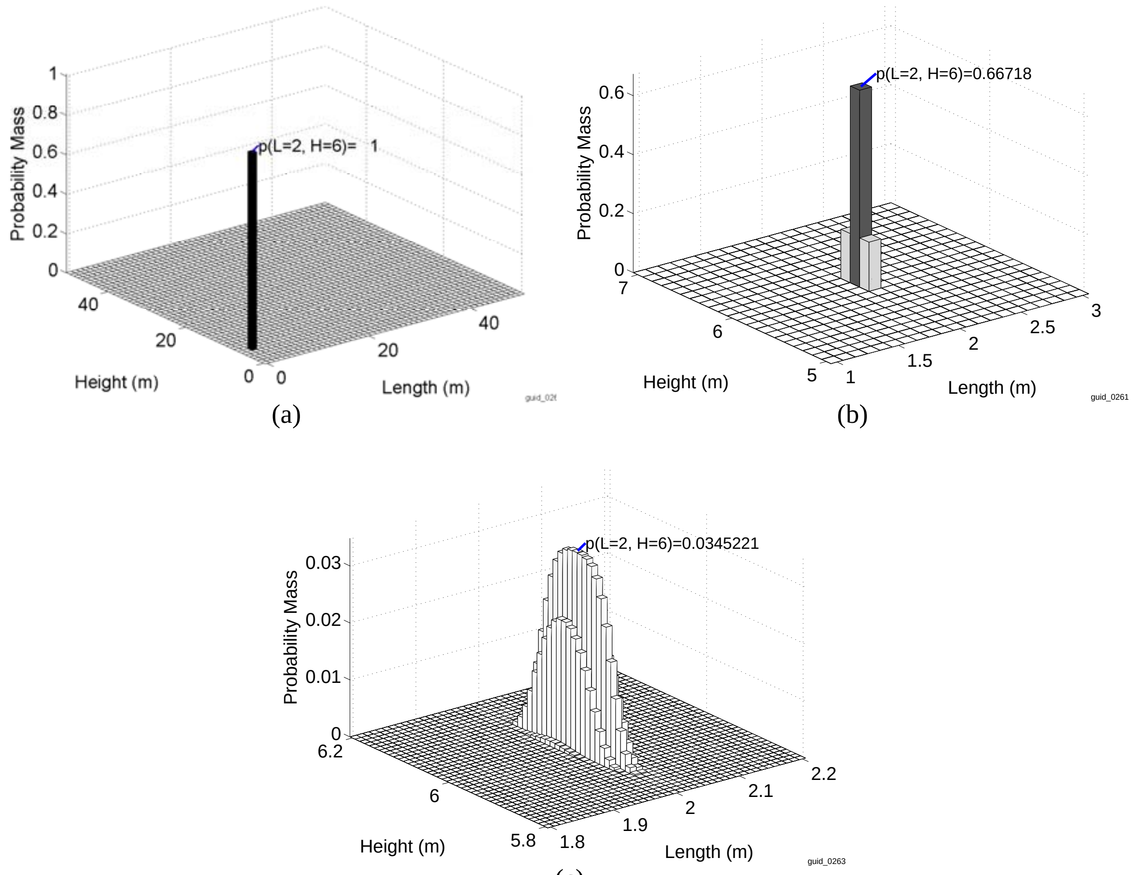
(a) (b)

(c) (d)

Figure 4.14: Estimation of the X and Y position a single dihedral shape located at position $X = 5, Y = 6$ at intervals of size (a) 5 m, (b) 1 m, (c) 0.1 m, and (d) 0.01m.

4.2.3.2 Dihedral Size Estimation.

This section tests the results of the estimator to determine size parameters (length and height). Figure 4.15 shows the results of progressively finer interval estimates. Again at first a coarse interval (1 m) and a large parameter space (L, H) of 50 m is used at first. The probability mass is concentrated at one point. The bounds are centered around this point in parameter space with a finer interval. The estimator consistently picks out the parameter. Again, the fine structure is evident at the increased zoom.



(a)

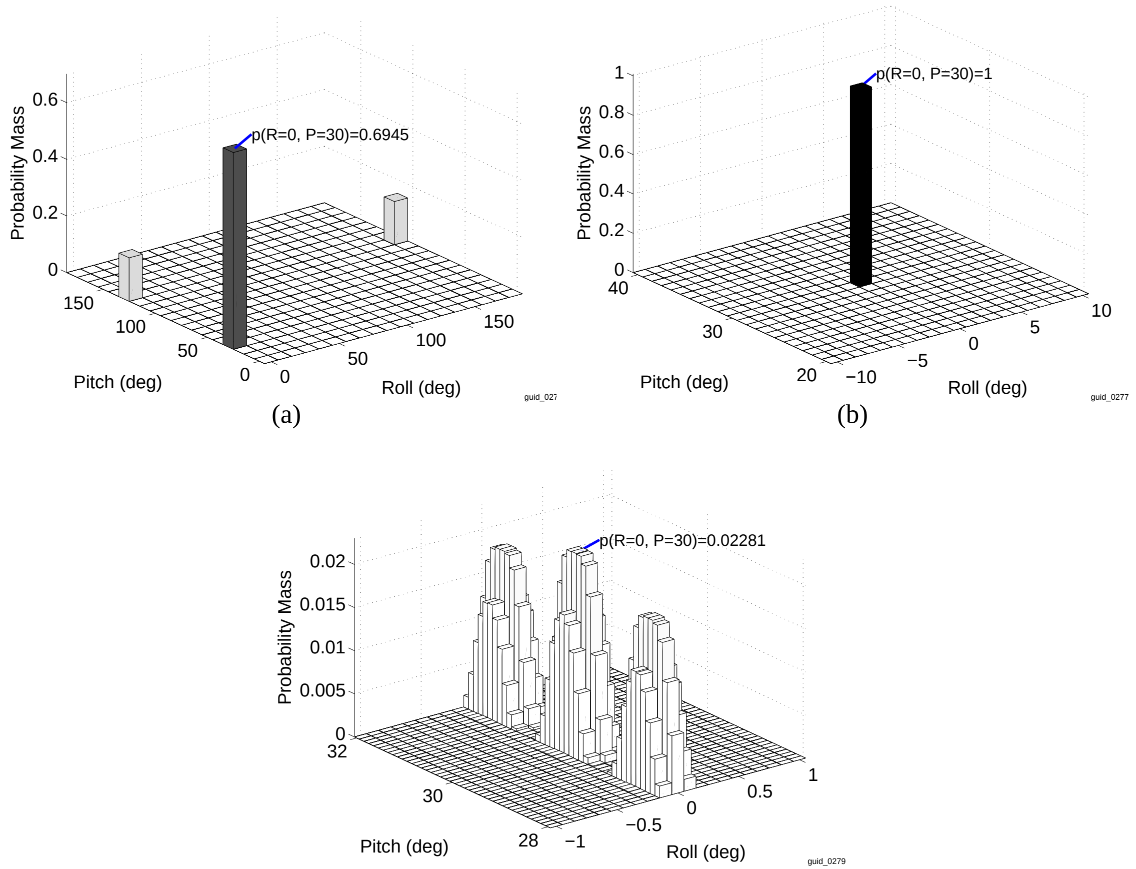
(b)

(c)

Figure 4.15: Estimation of the length and height parameters of a single dihedral shape of true size $L = 2$,$H = 6$ at intervals of size (a) 1m, (b)0.1m, and (c) 0.01m.

4.2.3.3 Dihedral Pose Estimation.

Figure 4.16 shows the results of successively finer intervals on estimation of the pose parameters roll and pitch (estimating yaw is similar). Again the fine structure is present at increased zoom. Note that the fine structure of this configuration is not evident until very small intervals as shown in Figure 4.16 (c). There are many local maxima around the true parameter values.



(a) (b)

(c)

Figure 4.16: Estimation of roll and pitch parameters for a dihedral at roll=0, pitch=30 at intervals of size (a) 10°, (b) 1°, and (c) 0.01°.

4.2.4 Single Trihedral Shape.

The interval estimation is performed over a series of known trihedral shapes. The trihedral shape has seven parameters: (X,Y,Z), (roll, pitch, yaw) and (height). The RCS area is calculated as $A = 2\sqrt{3}H^2$. At each simulation two shape parameters are estimated and the remaining are ‘given’ a-priori. This is equivalent to perfect knowledge of the other ‘nuisance’ parameters. Unless otherwise specified, the simulation parameters used for the ‘nuisance’ parameters are shown in Table 4.5.

Table 4.5: Single Trihedral Nuisance Parameters.

Name	**Value**	**Unit**
X	5	meters
Y	6	meters
Z	0	meters
Height	6	meters
RCS	$2\sqrt{3}H^2$	sq. meters
Roll	0	degrees
Pitch	30	degrees
Yaw	0	degrees
Polarization	VV	

4.2.4.1 *Trihedral Position Estimation.*

Figure 4.17 shows a series of progressively finer interval estimates of the position space. At first a very wide area of the (X, Y) parameter space (100 m) is evaluated using a coarse interval (5 m). This resolution is insufficient since the estimator finds no masses anywhere, so the resolution is increased to 1 m. A large probability mass is located near $(X, Y) = (5, 6)$. The interval is decreased as are the bounds of parameter space. In each step only the area of parameter space around the largest peak becomes the new bounds for the next finer interval. As the interval is increased, the fine structure of parameter space is revealed to be a highly concentrated probability mass at the true location.

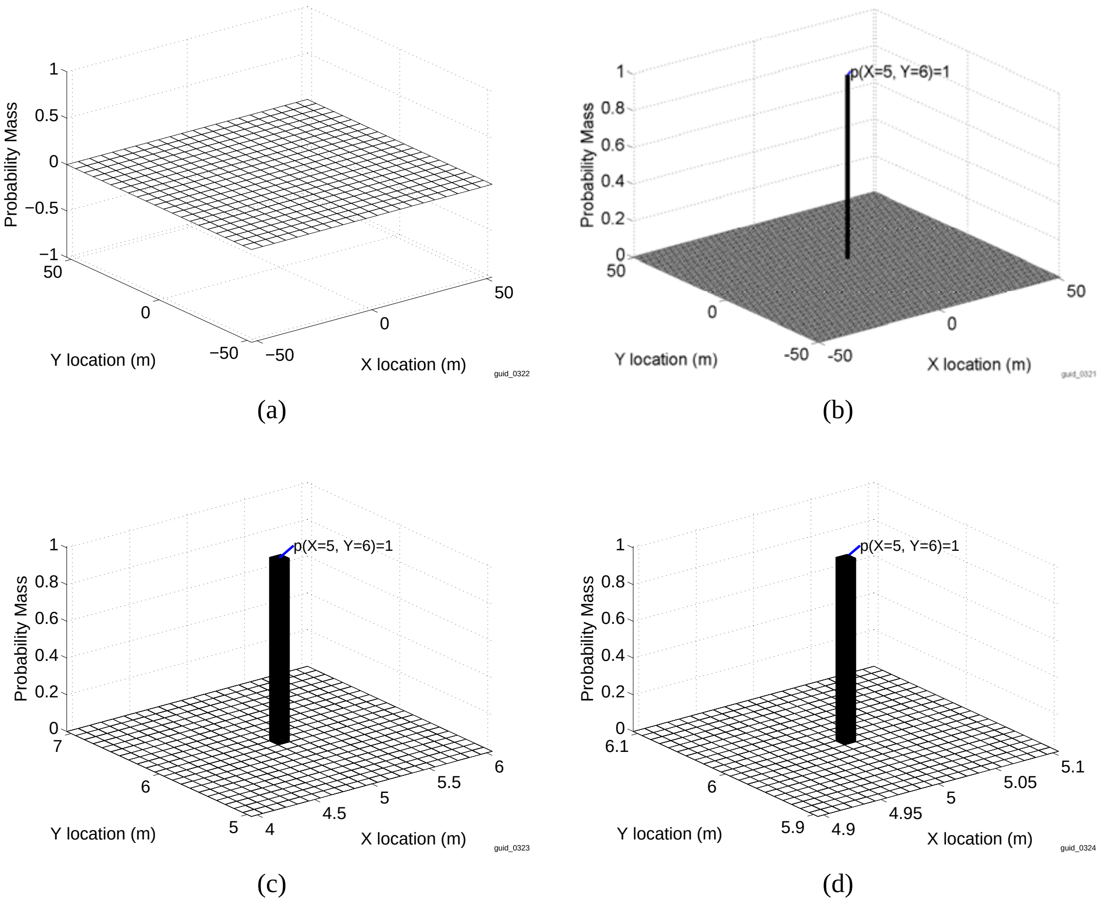
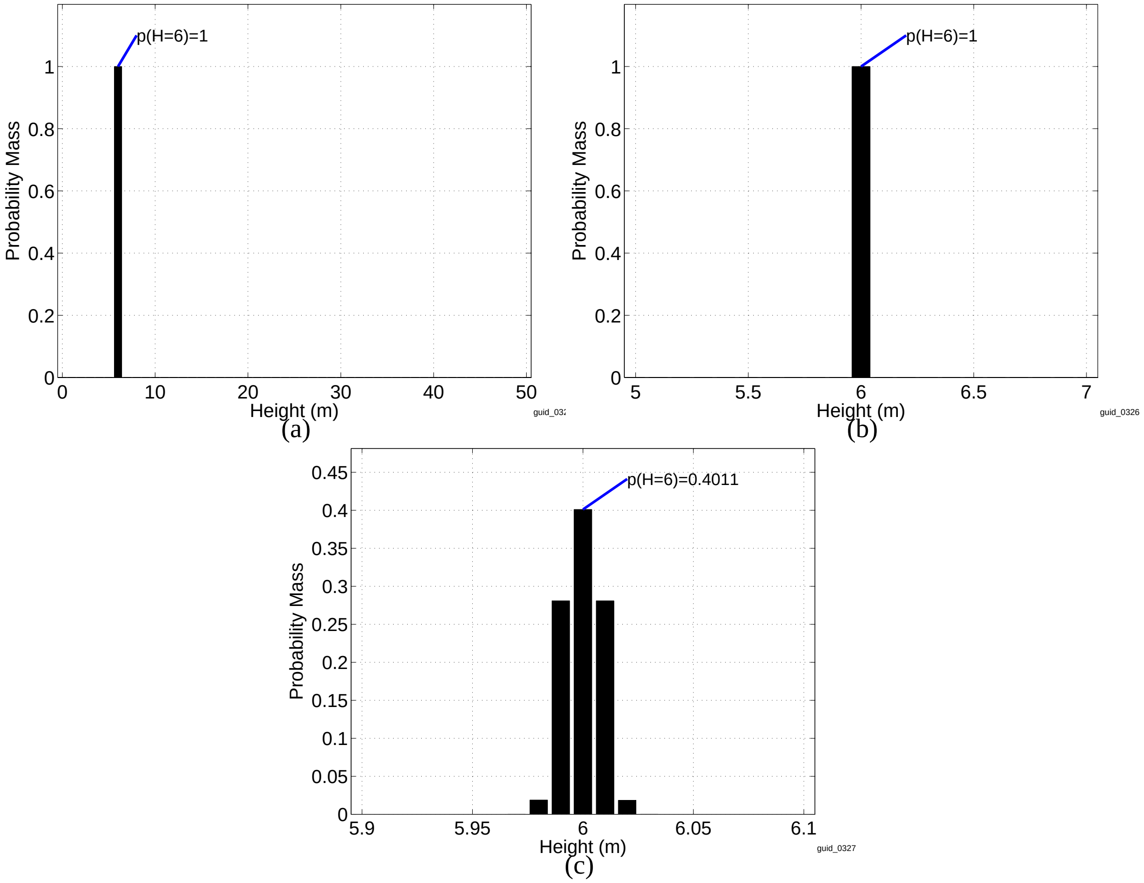


Figure 4.17: Estimation of the X and Y position a single trihedral shape located at position $X = 5, Y = 6$ at intervals of size (a) 5 m, (b) 1 m, (c) 0.1 m, and (d) 0.01m.

4.2.4.2 Trihedral Size Estimation.

This section tests the results of the estimator to determine size parameter (height). Figure 4.18 shows the results of progressively finer interval estimates. Again, a coarse interval (1 m) and a large parameter space H of 50 m are used initially. The probability mass is concentrated at one point. The bounds are centered around this point in parameter space with a finer interval. The estimator consistently picks out the parameter. Again, the fine structure is evident at the increased zoom.



(a)

(b)

(c)

Figure 4.18: Estimation of the height parameter of a single trihedral shape of true size $H = 6$ at intervals of size (a) 1m, (b) 0.1m, and (c) 0.01m.

4.2.4.3 Trihedral Pose Estimation.

Figure 4.19 shows the results of successively finer intervals on estimation of the pose parameters roll and pitch (estimating yaw is similar). Again the fine structure is present at increased zoom. Note that the fine structure of this configuration is not evident until very small intervals as shown in Figure 4.19 (c). There are many local maxima around the true parameter values.

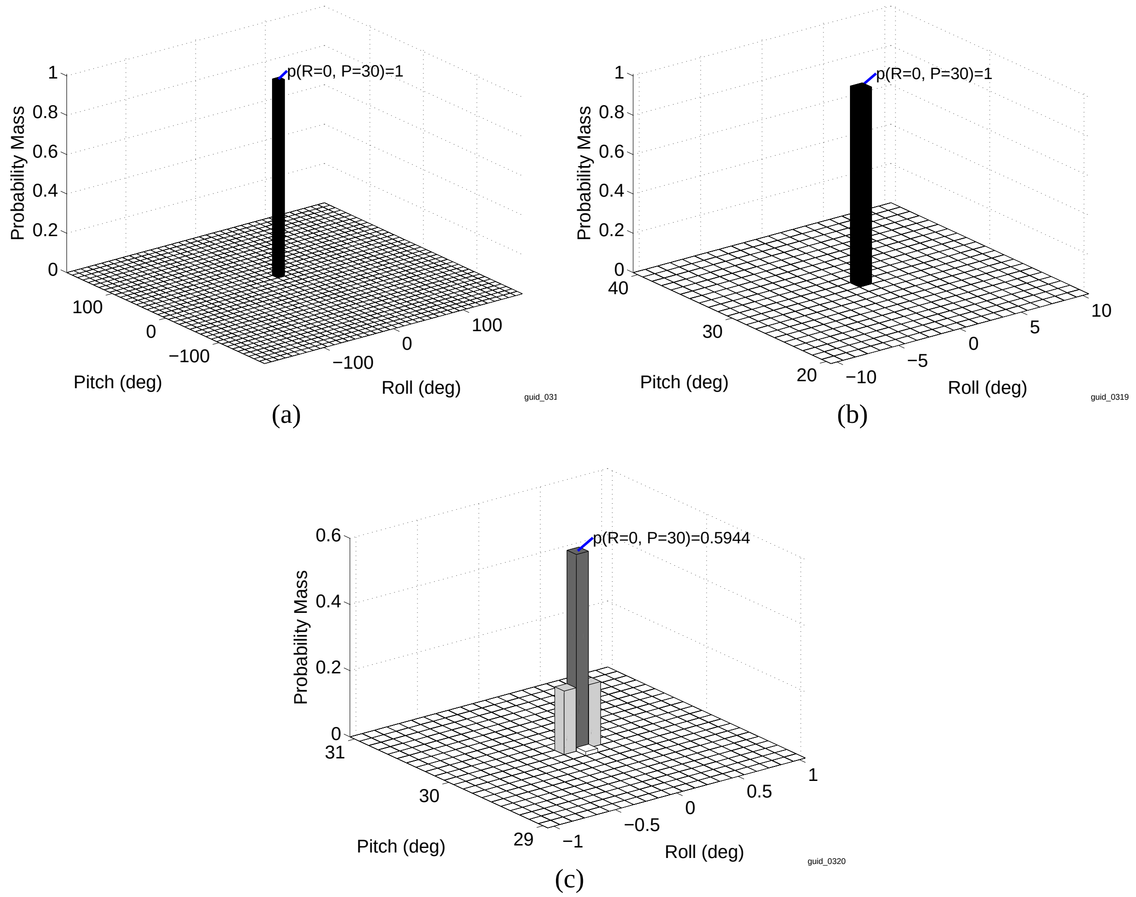


Figure 4.19: Estimation of roll and pitch parameters for a trihedral shape at roll=0, pitch=30 at intervals of size (a) 10°, (b) 1°, and (c) 0.01°.

4.2.5 Single Top-hat Shape.

The interval estimation is performed over a series of known top-hat shapes. The top-hat shape has eight parameters: (X,Y,Z), (roll, pitch, yaw) and (height, radius). The RCS area is calculated as $A = \sqrt{8R/\sqrt{2}} \times H$. At each simulation, two shape parameters are estimated and the remaining are ‘given’ a-priori. This is equivalent to perfect knowledge of the other ‘nuisance’ parameters. Unless otherwise specified, the simulation parameters used for the ‘nuisance’ parameters are shown in Table 4.6.

Table 4.6: Single Top-hat Nuisance Parameters.

Name	Value	Unit
X	5	meters
Y	6	meters
Z	0	meters
Radius	2	meters
Height	6	meters
RCS	$\sqrt{8R/\sqrt{2}} \times H$	sq. meters
Roll	0	degrees
Pitch	30	degrees
Yaw	0	degrees
Polarization	VV	

4.2.5.1 Top-hat Position Estimation.

Figure 4.20 shows a series of progressively finer interval estimates of the position space. At first, a very wide area of the (X, Y) parameter space (100 m) is evaluated using a coarse interval (5 m). This resolution is insufficient since the estimator finds no masses anywhere, so the resolution is increased to 1 m. A large probability mass is located near $(X, Y) = (5, 6)$. The interval is decreased as are the bounds of parameter space. In each step, only the area of parameter space around the largest peak becomes the new bounds for the next finer interval. As the interval is increased, the fine structure of parameter space is revealed to be a highly concentrated probability mass at the true location.

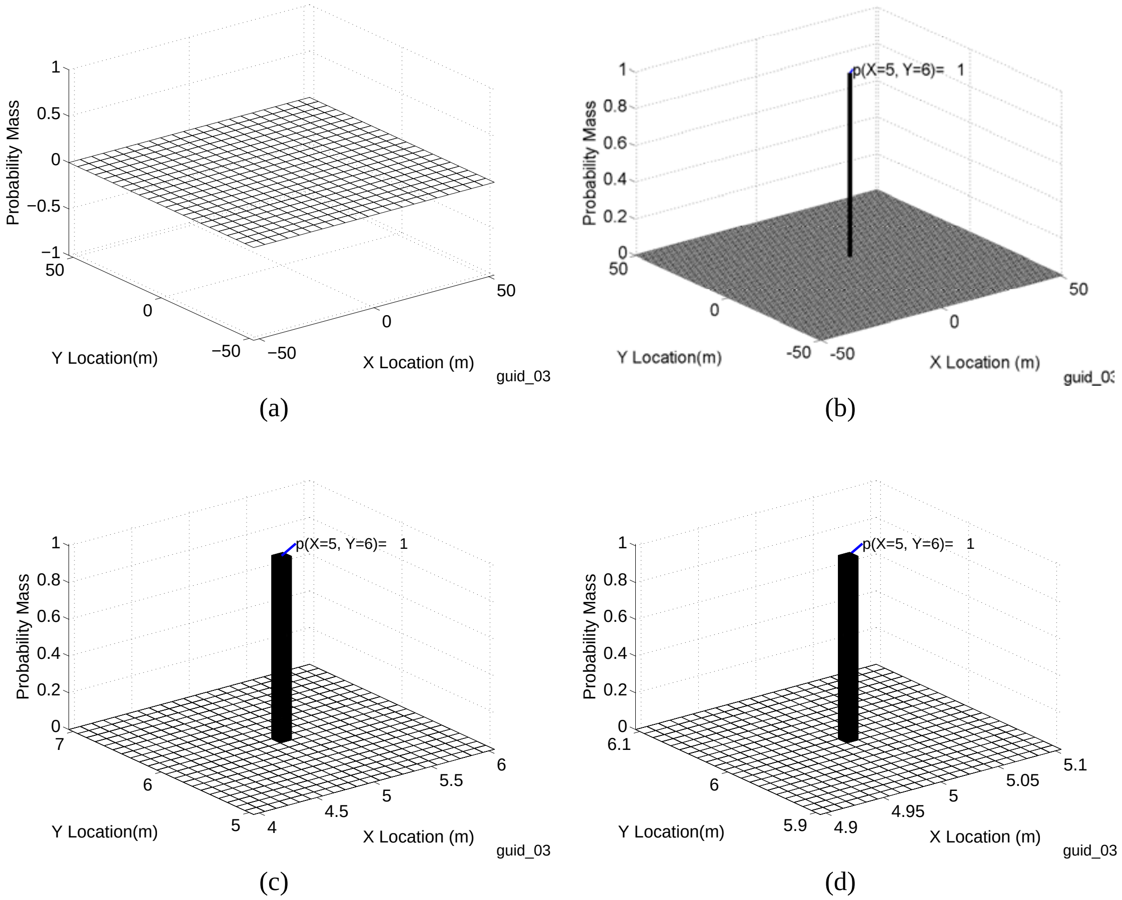


Figure 4.20: Estimation of the X and Y position a single top-hat shape located at position $X = 5, Y = 6$ over intervals of size (a) 5 m, (b) 1 m, (c) 0.1 m, and (d) 0.01 m.

4.2.5.2 Top-hat Size Estimation.

This section tests the results of the estimator to determine size parameters (radius and height). Figure 4.21 shows the results of progressively finer interval estimates. Again, at first a coarse interval (1 m) and a large parameter space (R, H) of 50 m is used. The probability mass is concentrated at one point. The bounds are centered around this point in parameter space with a finer interval. The estimator consistently picks out the parameter. Again, the fine structure is evident at the increased zoom.

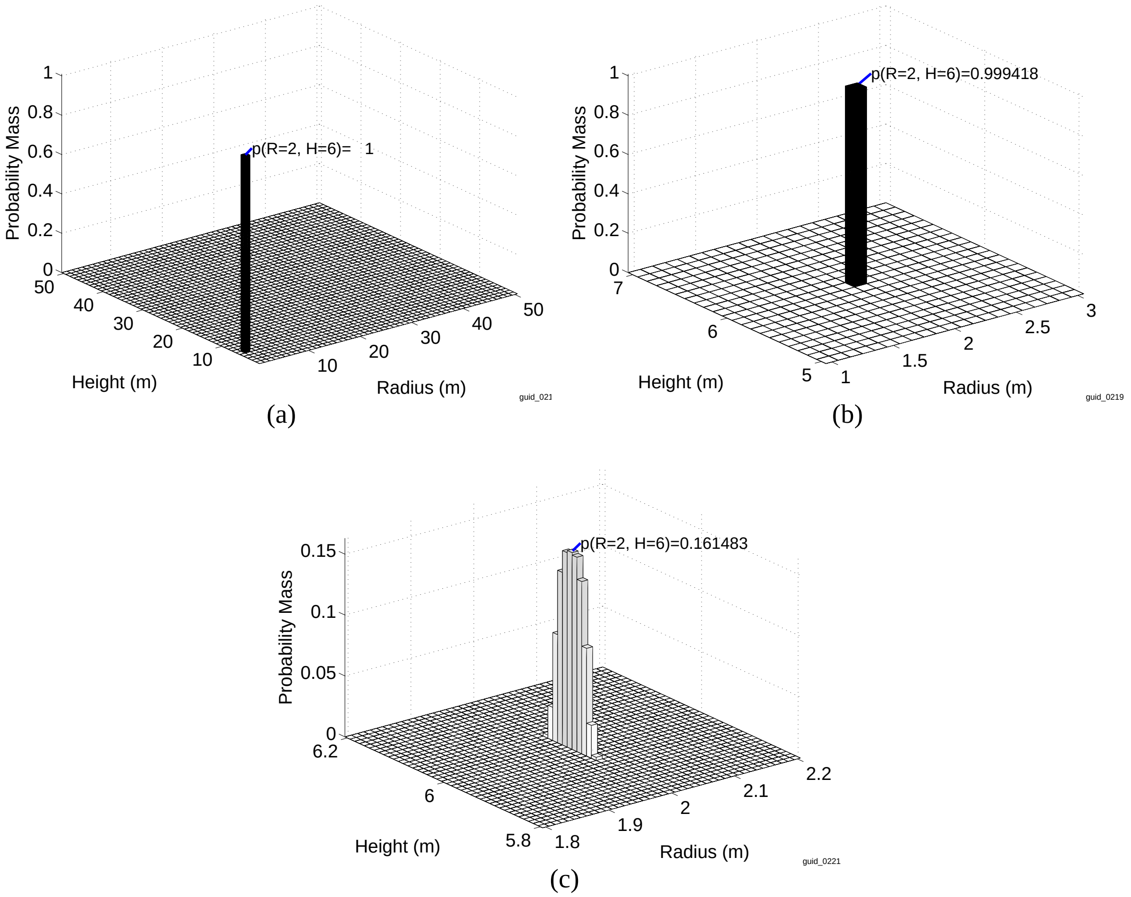
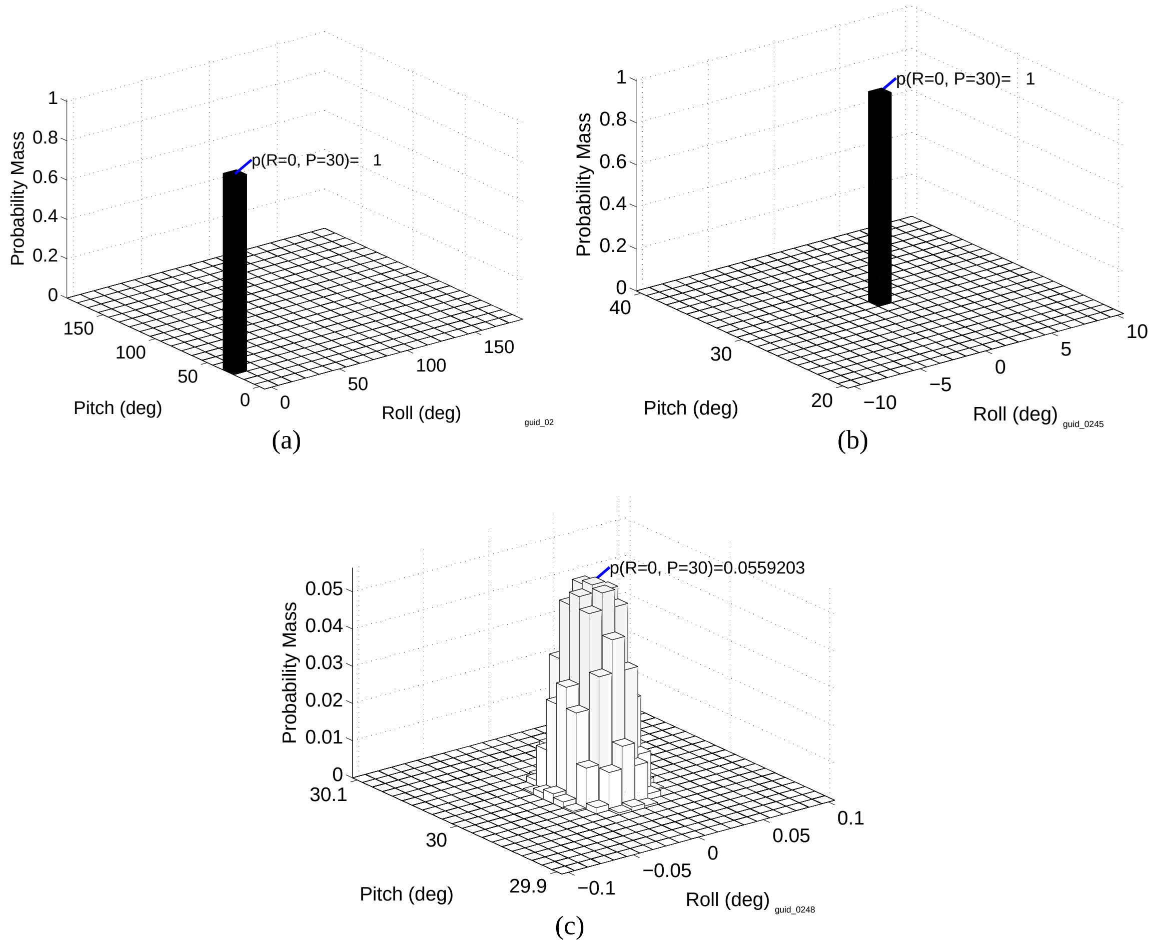


Figure 4.21: Estimation of the radius and height parameters of a single top-hat shape of true size $R = 2$, $H = 6$ at intervals of size (a) 1m, (b) 0.1m, and (c) 0.01m.

4.2.5.3 Top-hat Pose Estimation.

Figure 4.22 shows the results of successively finer intervals on estimation of the pose parameters roll and pitch (estimating yaw is similar). Again the fine structure is present at increased zoom. Note that the fine structure of this configuration is not evident until very small intervals as shown in Figure 4.22 (c). The probability mass is centered in an area of $\pm 0.03^\circ$, but the estimator consistently finds a peak even at the low resolution of 10°.



(a)

(b)

(c)

Figure 4.22: Estimation of roll and pitch parameters for a top-hat at roll=0, pitch=30 at intervals of size (a) 10°, (b) 1°, and (c) 0.01°.

4.2.6 Single Cylinder Shape.

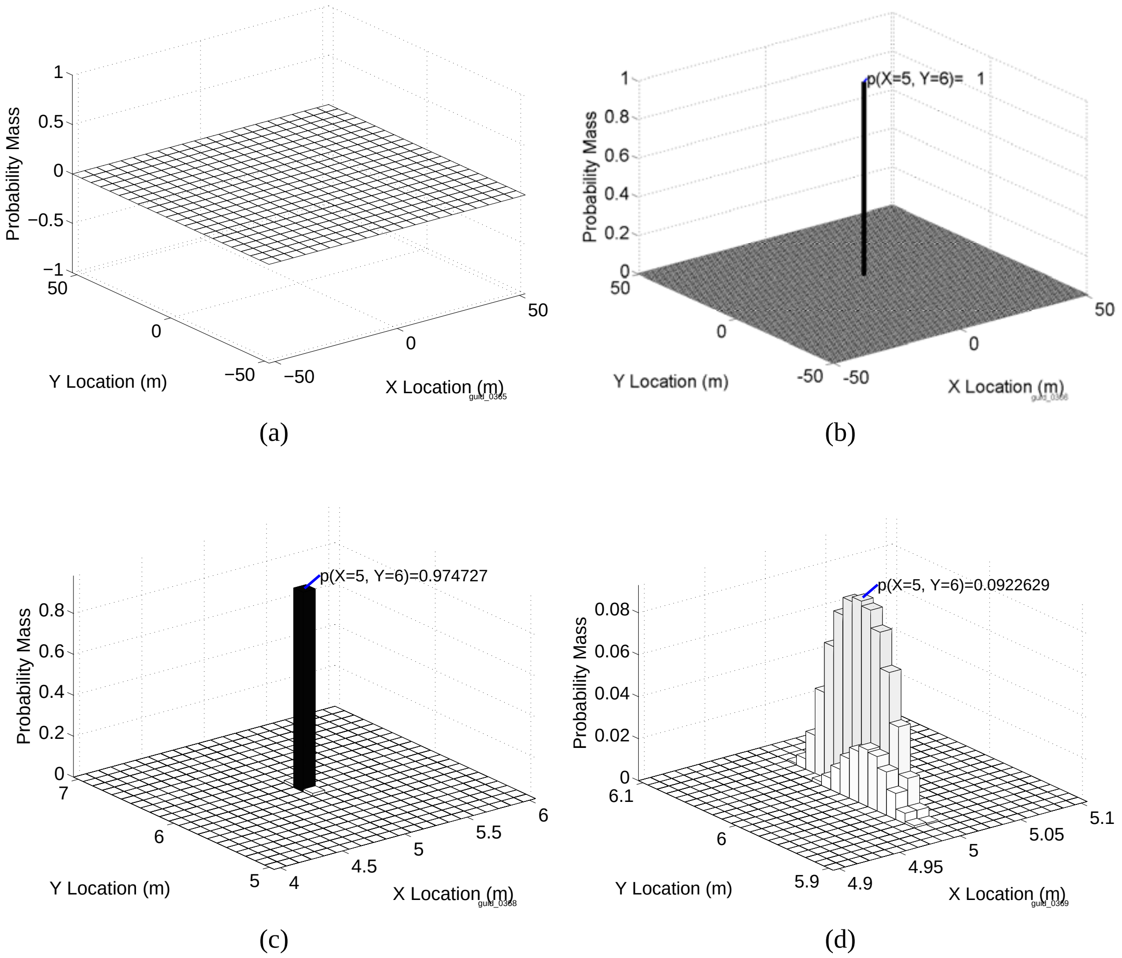
The interval estimation is performed over a series of known cylinder shapes. The cylinder shape has eight parameters: (X,Y,Z), (roll, pitch, yaw) and (length, radius). The RCS area is calculated as $A = L\sqrt{R}$. At each simulation, two shape parameters are estimated and the remaining are 'given' a-priori. This is equivalent to perfect knowledge of the other 'nuisance' parameters. Unless otherwise specified, the simulation parameters used for the 'nuisance' parameters are shown in Table 4.7.

Table 4.7: Single Cylinder Nuisance Parameters.

Name	**Value**	**Unit**
X	5	meters
Y	6	meters
Z	0	meters
Radius	2	meters
Height	6	meters
RCS	$L\sqrt{R}$	sq. meters
Roll	5	degrees
Pitch	30	degrees
Yaw	0	degrees
Polarization	VV	

4.2.6.1 Cylinder Position Estimation.

Figure 4.23 shows a series of progressively finer interval estimates of the position space. At first a very wide area of the (X, Y) parameter space (100 m) is evaluated using a coarse interval (5 m). This resolution is insufficient since the estimator finds no masses anywhere, so the resolution is increased to 1 m. A large probability mass is located near $(X, Y) = (5, 6)$. The interval is decreased as are the bounds of parameter space. In each step only the area of parameter space around the largest peak becomes the new bounds for the next finer interval. As the interval is increased, the fine structure of parameter space is revealed.



(a) (b)

(c) (d)

Figure 4.23: Estimation of the X and Y position a single cylinder shape located at position $X = 5, Y = 6$ at intervals of size (a) 5 m, (b) 1 m, (c) 0.1 m, and (d) 0.01m.

4.2.6.2 Cylinder Size Estimation.

This section tests the results of the estimator to determine size parameters (radius and length). Figure 4.24 shows the results of progressively finer interval estimates. Again at first a coarse interval (1 m) and a large parameter space (R, H) of 50 m is used at first. The probability mass is concentrated at one point. The bounds are centered around this point in parameter space with a finer interval. The estimator consistently picks out the parameter. Again, the fine structure is evident at the increased zoom.

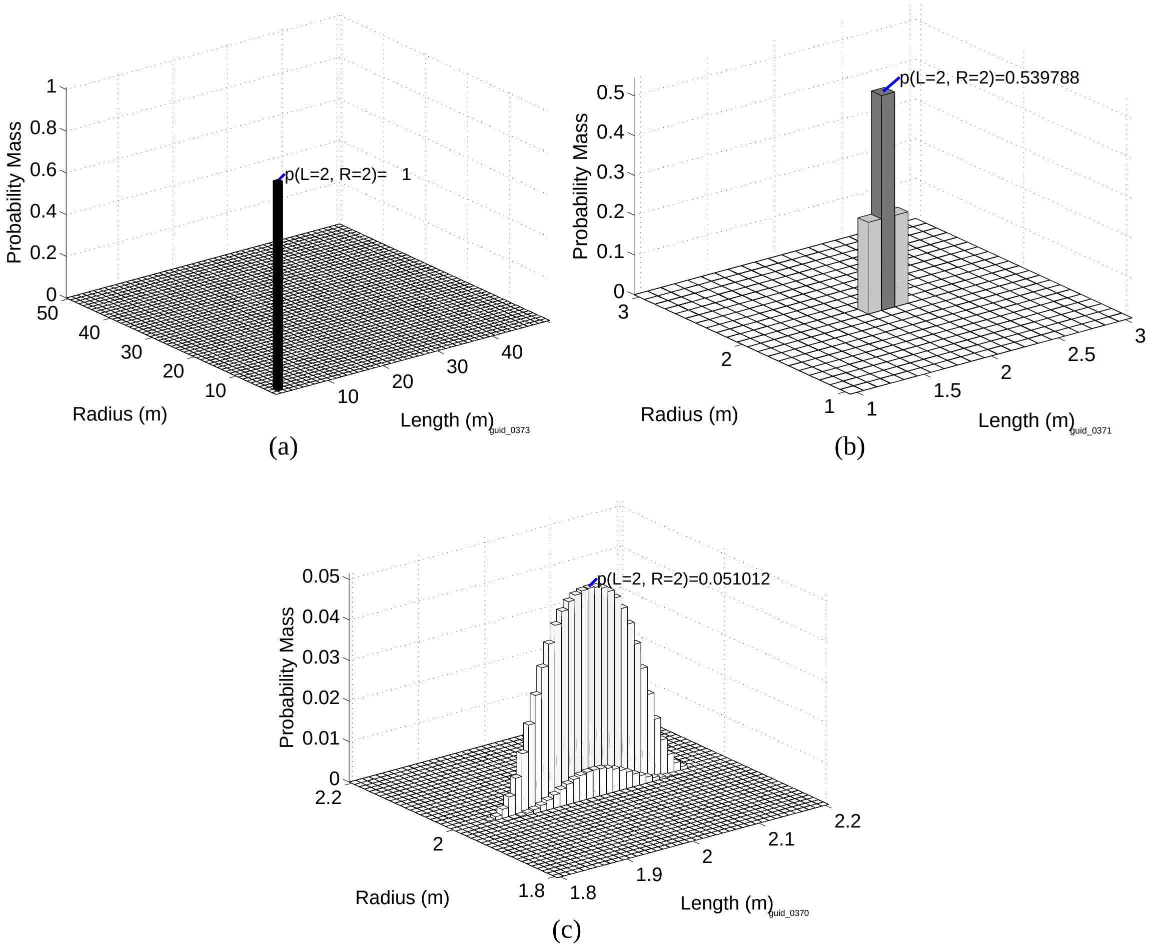
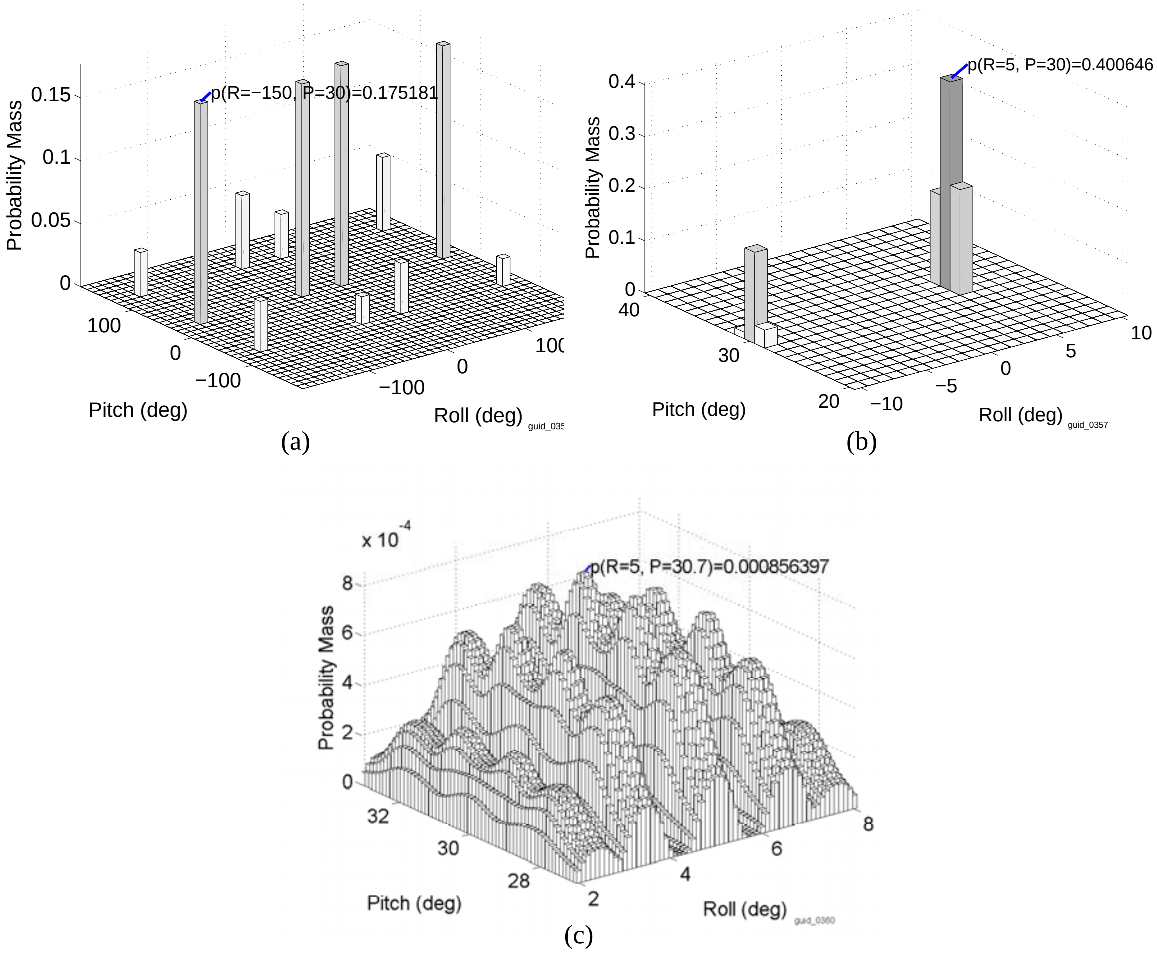


Figure 4.24: Estimation of the radius and height parameters of a single cylinder shape of true size $R = 2, L = 2$ at intervals of size (a) 1m, (b)0.1m, and (c) 0.01m.

4.2.6.3 Cylinder Pose Estimation.

Figure 4.25 shows the results of successively finer intervals on estimation of the pose parameters roll and pitch (estimating yaw is similar). Again, the fine structure is present at increased zoom. This shape suffers from the same periodicity in pose parameter that is seen in the plate shape. Note that the fine structure of this configuration is not evident until very small intervals as shown in Figure 4.25 (c). The probability mass is centered in an area of ±3°, but the estimator consistently finds a peak even at the low resolution of 10°. In Figure 4.25(c) the fine structure is much more complicated than originally indicated at low resolution. The peak of the estimate at 0.1° is slightly off. It is believed that there are actually multiple peaks of exactly the same mass and the estimator picked the first.



(a)

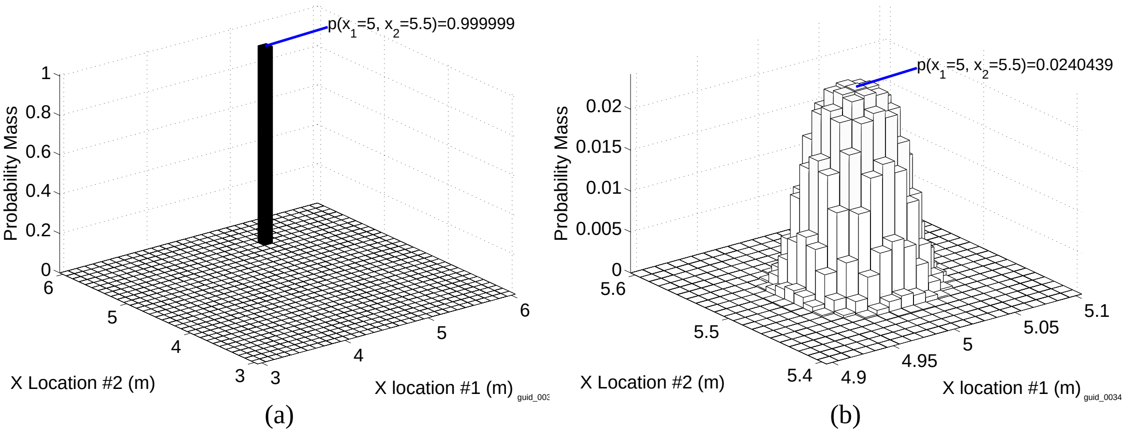
(b)

(c)

Figure 4.25: Estimation of roll and pitch parameters for a cylinder at roll=5, pitch=30 at intervals of size (a) 10°, (b) 1°, and (c) 0.1°.

4.3 Two Shape Case

All target scenes of interest have multiple shapes of various parameters. From the parameter estimation case, the multiple shapes simply increase the dimensionality of the parameter space $\underline{\mathbf{\Theta}}$. For N plate shapes in the scene, each having 8 parameters the parameter space is $8N$ dimensions. Figure 4.26 shows the interval estimates for a phase history that is the sum of two plates. In this case the other 14 parameters are ‘given’, and identical except that shape 1 has $Y_1 = 5$ and shape 2 has $Y_2 = 3$. Note that the familiar single mass peak is seen at coarse intervals with its fine structure visible as interval is decreased.

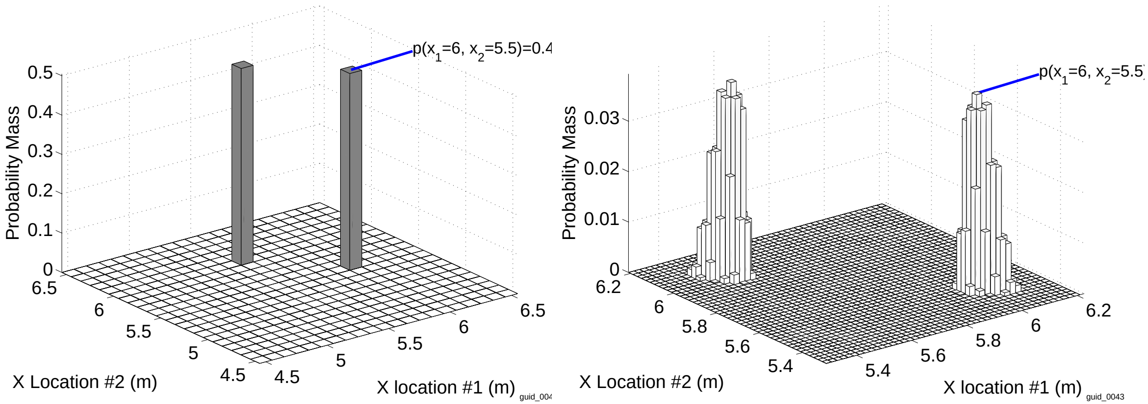


(a) (b)

Figure 4.26: Estimation of X position of two plates (different Y) located at $X_1 = 5, X_2 = 5.5$ at intervals of size (a) 0.1 m, and (b) at 0.01m.

Figure 4.27 shows when the two shapes have *all* ‘given’ parameters identical. In this case there is no distinction between the labeling of shapes 1 and 2. With no given parameter to establish labeling, there appear two mass peaks for the two permutations of X location. This is an example where the assumption of the statistical independence of parameters is broken.

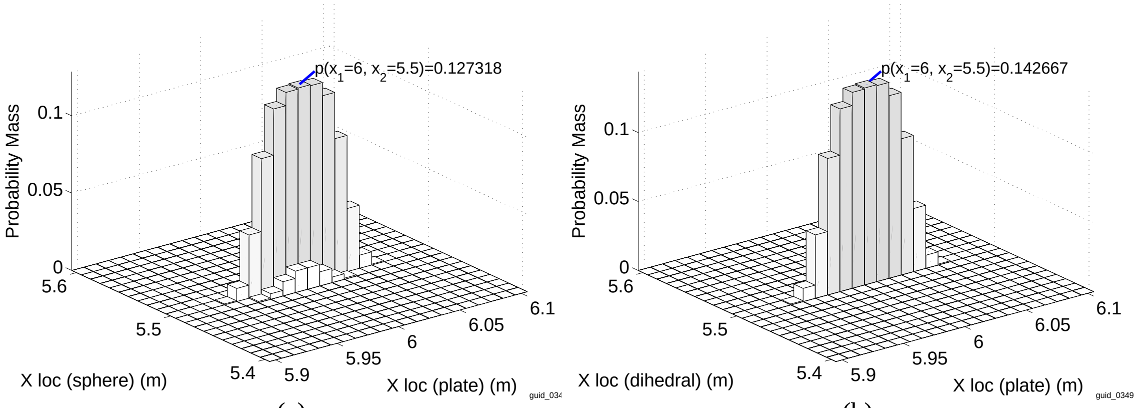
To study the effects of the estimator for two dissimilar shapes, we performed a series of simulations with parameters similar to that of the two plate case but for a phase history containing both a plate and a sphere. In Figure 4.28 plot (a), the plate is located at position $(6, 5, 0)$ with $L = 2, H = 6$. The sphere is located at $(5.5, 5, 0)$ with $R = 1.5$. In Figure 4.28(b) The scene is for a plate at $(6, 5, 0)$ with $L = 6, H = 2$ and a dihedral at



(a) (b)

Figure 4.27: Estimation of X Position of two plates (otherwise identical) located at $X_1 = 6$, $X_2 = 5.5$ over intervals of size (a) 0.1 m, and (b) 0.02m.

(5.5, 5, 0) with $L = 6, H = 2$. The two shapes are clearly present and their coordinates are estimated accurately. Note that the change in the second shape slightly changes the width of the credible interval (viewed as the width of the 'bump'), while the credible interval of the plate axis is the same.



(a) (b)

Figure 4.28: X position estimates of two different shapes present together: (a) plate and sphere, and (b) plate and dihedral.

4.4 Incorrect Shape Type

This section discusses the effects of using the wrong shape type in an estimation problem. The wrong shape scenario could happen if the shape type is not known, and the user tries to estimate the parameters of each shape type based on a given observation $\underline{\boldsymbol{y}}$. A phase history of each shape type is subjected to the estimators of the other shape type. For these experiments, we limited the estimation only to the size parameters (length, height, and radius as appropriate).

In most cases, incorrect shape assumption creates a situation where the likelihood function $\boldsymbol{p}(\underline{\boldsymbol{y}} \mid \underline{\boldsymbol{\Theta}})$ is extremely small. In the cases studied during this research, the likelihood function is below the numerical limits of the computing platform. Table 4.8 summarizes the results of each permutation. For some combinations, the estimator is able to create a set of probability masses. In those cases we have generated figures of the estimator output for study.

Table 4.8: Results of incorrect shape type assumption. "Low" indicates uniformly low probability of parameter over the whole space.

		True Shape					
		Plate	Dihedral	Trihedral	Sphere	Top-hat	Cylinder
Assumed	Plate	—	Low	Low	Low	Low	Low
	Dihedral	$L = 0, H = 0$	—	Low	Low	Low	Low
	Trihedral	Low	Low	—	Low	Low	Low
	Sphere	Low	$R = 0$	$R = 0$	—	Low	R=0
	Top-hat	$H = 0, R = any$	Low	Low	Low	—	Low
	Cylinder	$R = 0$	Low	Low	Low	Low	—

Figure 4.29 shows the results of attempting to estimate the radius parameter assuming the phase history is that of a sphere. In Figure 4.29(a) the true shape is a 2×6 dihedral and the estimator sees this as a zero-radius sphere. In Figure 4.29(b) the true shape is a 2×6 plate and the estimator sees this again as as a zero-radius sphere. In Figure 4.29(c) the true shape is a $R = 1, L = 2$ cylinder.

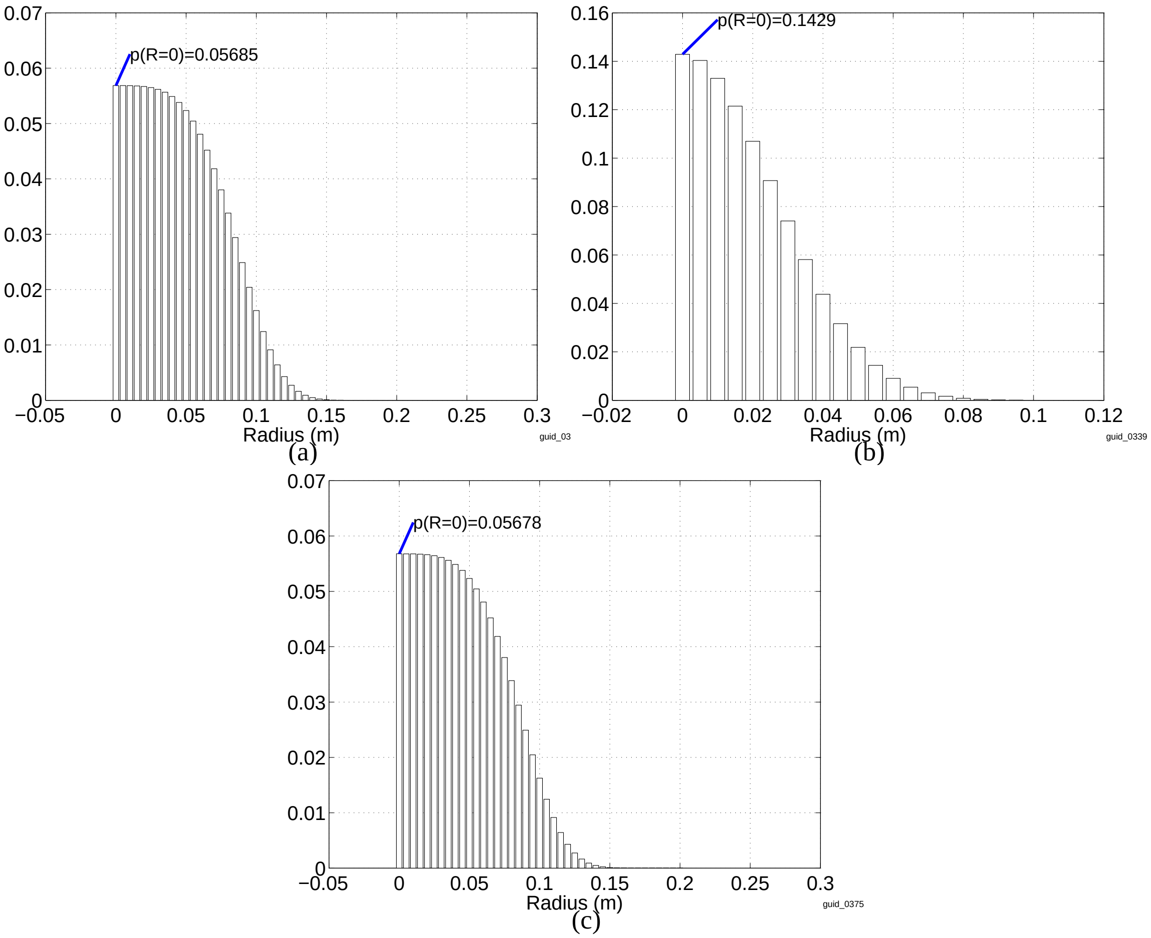


Figure 4.29: Estimating a sphere with the wrong shape when the true shape is (a) a dihedral, (b) a plate, and (c) a cylinder.

Figure 4.30 shows the results of attempting to estimate the height and length parameters assuming the phase history is that of a dihedral. When the plate phase history is applied to this estimator, the result is a line of probability masses along the $H = 0$ and $L = 0$ ordinates.

Figure 4.31 shows the results of attempting to estimate the height and radius parameters assuming the phase history is that of a top-hat when the true phase history is a plate. When the plate phase history is applied to this estimator the result is a line of probability masses along the $H = 0$ ordinate only.

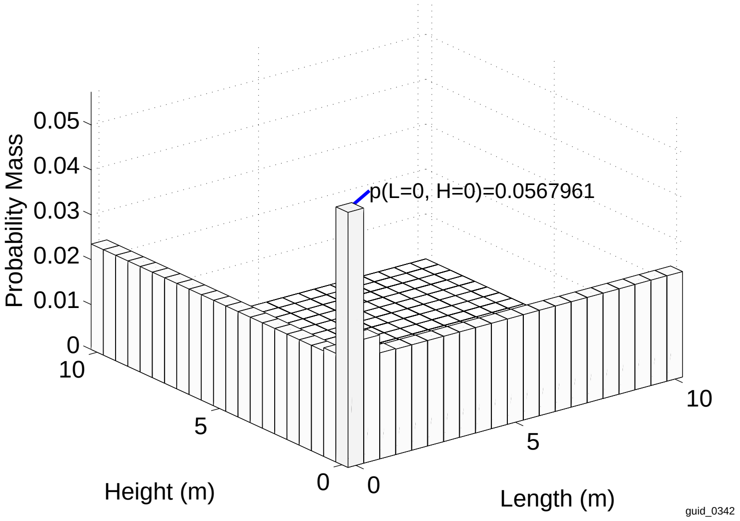


Figure 4.30: Estimating a dihedral with the wrong shape when the true shape is a plate.

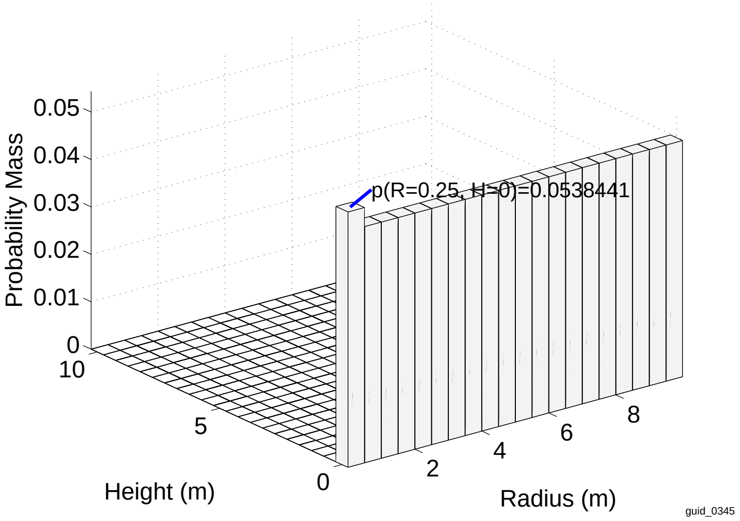


Figure 4.31: Estimating a top-hat with the wrong shape when the true shape is a plate.

Figure 4.32 shows the results of attempting to estimate the length and radius parameters assuming the phase history is that of a cylinder when the true phase history is a plate. When the phase history of a plate is applied to this estimator the result is a line of probability masses along the $R = 0$ ordinate only.

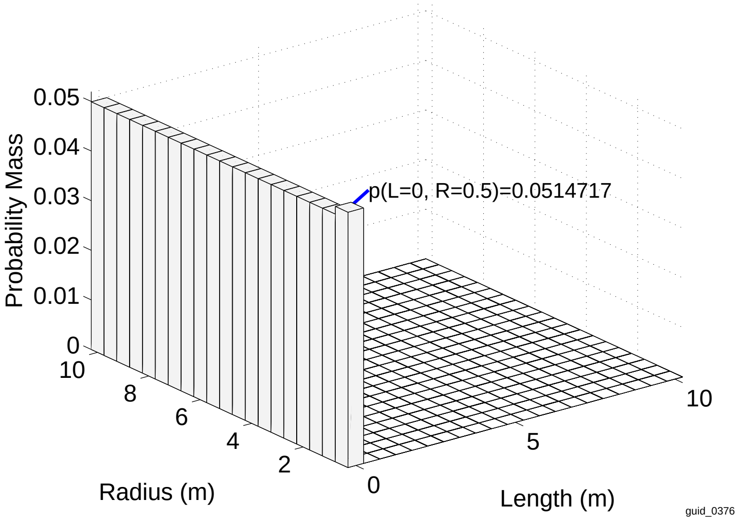


Figure 4.32: Estimating a cylinder with the wrong shape when the true shape is a plate.

The case of incorrect shape assumption often yield in one of two results: a uniformly "low" probability mass, or a distribution of masses along a zero-size axis. Measuring low probability mass is easy for practical implementations to check for, and judicious use of the prior distribution $\boldsymbol{p}(\underline{\boldsymbol{\Theta}})$ can remove degenerate cases like those seen in Figures 4.31-4.32.

4.5 Computation Time

The plots contained in this chapter are generated using 20 × 20 probability mass intervals. For these 400 masses, the parameters of Table 4.1 require $F = 64$ frequency bins, $K = 181$ azimuth/elevation points and $N = 5$ order polynomial. For the plate, there are $D = 8$ parameter dimensions. Using the results of Section 2.6 each sample requires 9.05 billion calculations of the magnitude and phase factors in the Jackson model to calculate each probability mass [2]. Prior to GPU implementation this calculation took about 13 hours on a quad-core processor. After implementing the Jackson models and associated code into the CUDA GPU the computations were reduced to 12 minutes.

In the figures generated for this chapter, we constrained the problem to only $D = 2$ parameters, resulting in 579,200 calculations per mass, or 116 million calculations for the figure. An early baseline of the $D = 2$ estimation of length and height of a plate prior to CUDA implementation took approximately 1.3 hours per figure. Using the CUDA GPU implementation of the Jackson models yields a computation time of about 5-10 seconds per figure.

4.6 Credible Regions

Section 3.4.1 describes the calculation of measures of confidence through the use of the credible set C_P. In this section, we demonstrate the results of a calculation of credible set for the uninformative prior. Figure 4.33 shows the credible set C_P for the two-plate case of Section 4.3. The phase history is that used to generate Figure 4.26, where there are two plates in the target scene with different X and Y locations (pose and size parameters are the same for each shape). As in Section 4.3, the Y locations are 'given' as $Y_1 = 3$ and $Y_2 = 5$; only the X positions X_1 and X_2 are estimated. Figure 4.33(a) shows the probability mass at an interval of 0.01 m for two plates. Figure 4.33(b) shows the results of simple thresholding; all intervals with mass less than $\gamma = 0.001$ are excluded from the set. In Figure 4.33(c) we see the results of applying the ordering operator $\boldsymbol{O}[P]$ and the resulting sequence of probability masses. Finally, Figure 4.33(d) shows the result of applying only the M-largest masses such that the parameter $\alpha \geq 0.95$.

In the above figures, Figure 4.33(b) and (d) appear very similar, however different methods are used to determine if a particular interval is included in the credible set or

[2] These reference calculations of the entire magnitude and phase factors given in Chapter 2. Each calculation of $\underline{\boldsymbol{S}}(k)$ requires many thousand calculations by itself.

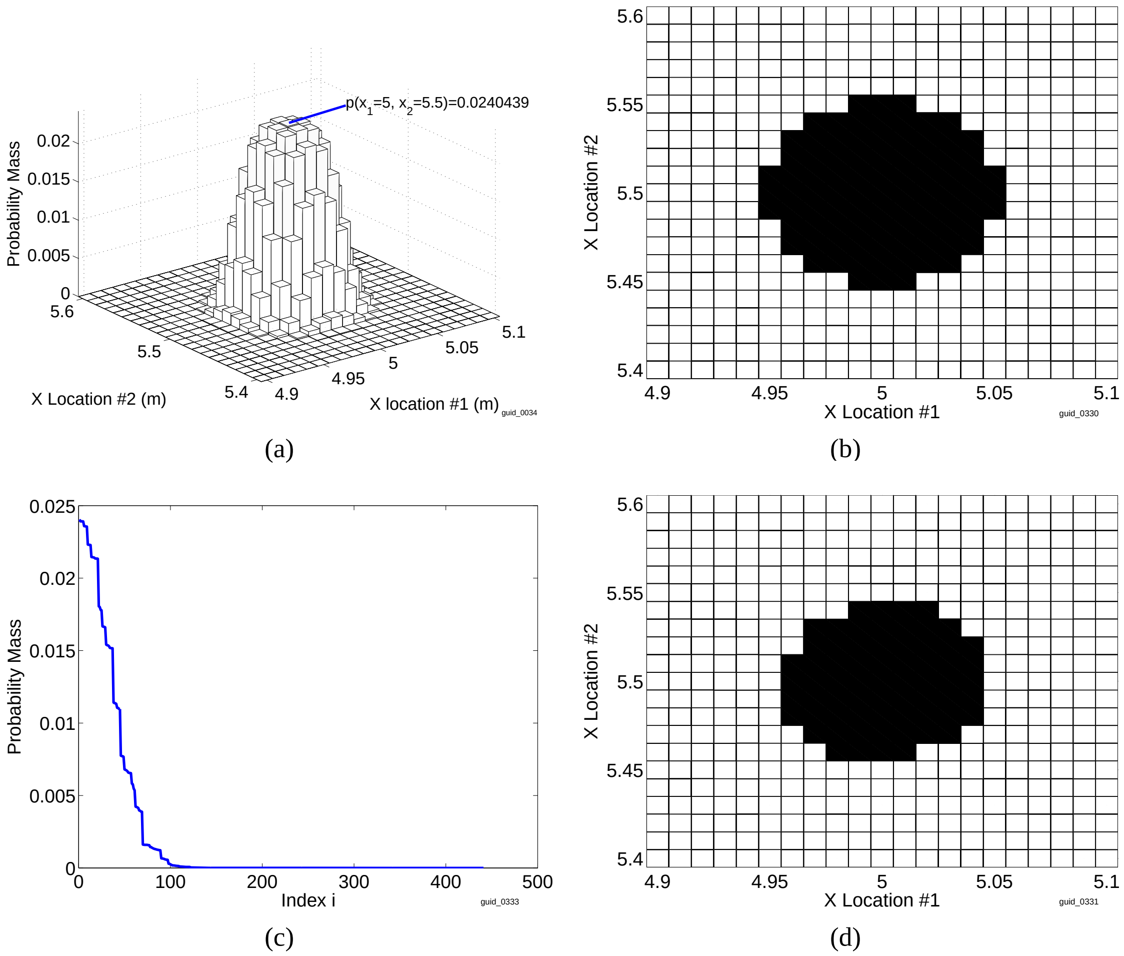
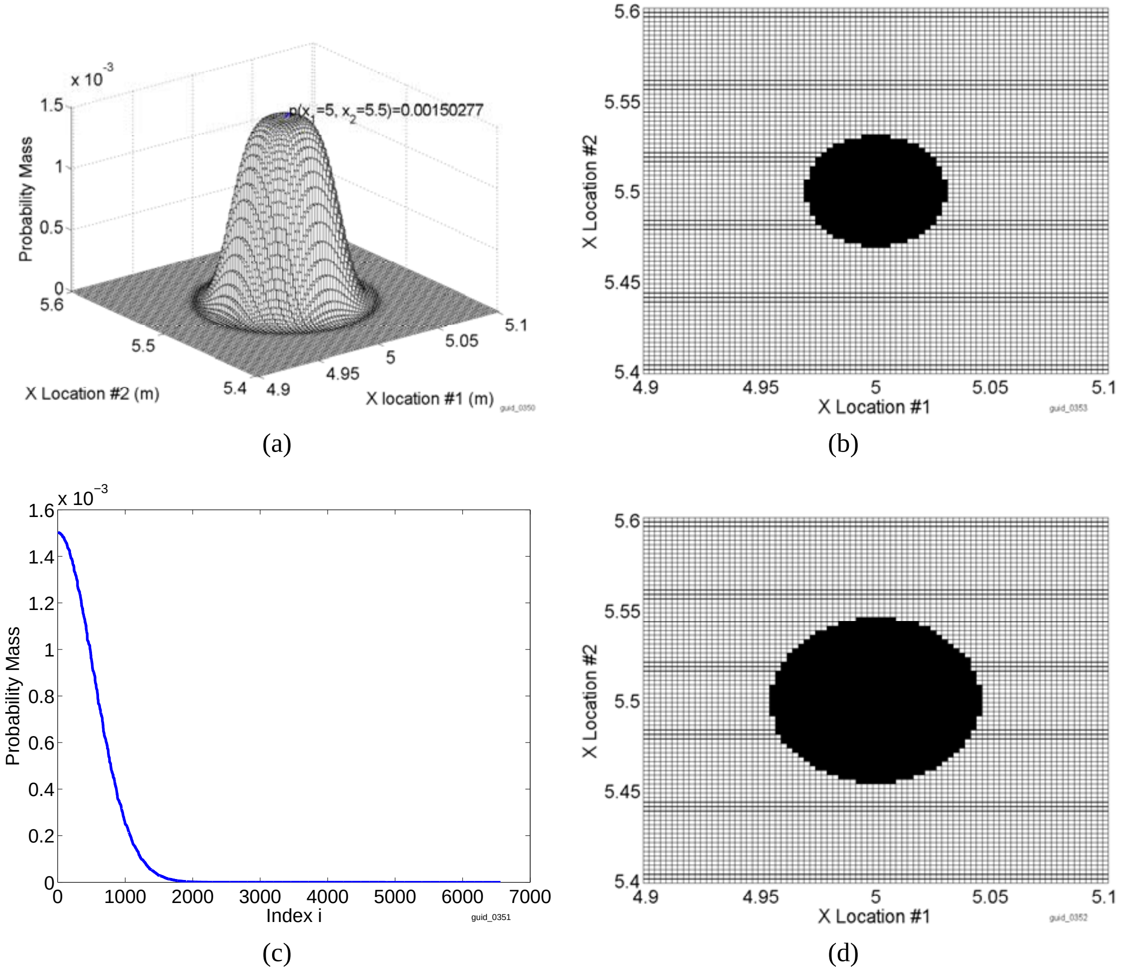


Figure 4.33: Credible set for two plates at $X_1 = 5$, $X_2 = 5.5$ (with $Y_1 = 3$, $Y_2 = 5$). (a) Probability mass for 0.01 m interval, (b) thresholded at $\gamma = 0.001$, (c) masses ordered highest to lowest, and (d) Upper credible interval C_P^+, α=0.95.

not. In Figure 4.33(b), the user must specify the threshold γ. For this plot, $\gamma = 0.001$ is defined by visual inspection of the probability masses in Figure 4.33(a). The value of γ must be recalculated whenever the interval size changes. However, in Figure 4.33(d) the user need only specify the parameter α. The number M of ordered masses that sum to α is automatically recalculated based on the probability mass set P that is given.



(a)

(b)

(c)

(d)

Figure 4.34: Credible set for two plates at $X_1 = 5$, $X_2 = 5.5$ (with $Y_1 = 3$, $Y_2 = 5$). (a) Probability mass for 0.0025 m interval, (b) thresholded at $\gamma = 0.001$, (c) masses ordered highest to lowest, and (d) upper credible interval C_P^+, α=0.95.

To display the effects of constant mass calculation versus simple thresholding, a new estimate of the probability mass with smaller intervals is shown in Figure 4.34. The interval is one-fourth the size of Figure 4.33. With the increased fine structure, we expect the credible region size not to change appreciably (although increased detail is expected). As shown in plot (a) of both Figure 4.33 and Figure 4.34, the probability mass set has more members at finer detail, and the mass values are smaller (being integrals of the same function over a smaller region). Plot (b) of each shows the effect of keeping the same threshold $\gamma = 0.001$. The bounds of the region actually decrease because γ has remained the same. In contrast, plot (d) of each figure shows the region bounds stay largely the same despite the fact that α remained fixed at 0.95, indicating the credible region is fairly consistent across these two interval sizes.

V. Discussion and Future Work

This research has focused on the detection, estimation and confidence determination of the Jackson canonical shapes for synthetic aperture radar (SAR).

Over the course of this thesis, we presented the background on the Jackson phase history models. We discussed how these models are used to the problem of this research: the detection of canonical shapes in phase history. We have developed an analysis of the posterior density space by means of statistical methods and numerical techniques. The results are promising, and in simulations we were able to find estimates of the true parameters in most cases, except where numerical precision became a problem.

The estimation techniques we have developed are applicable to any Bayesian estimation problem. The interval estimation is particularly useful in situations where a parameter estimate is needed only to a prescribed precision, such as plus-or-minus a certain tolerance. The multi-zoom capability allows the user to terminate the estimate when the desired precision is achieved. It also allows a ‘quick look’ at posterior space as well as detailed views. The phase history model we used was that of Jackson [1], however the probability mass technique can be used for other models, such as the traditional point scatterer model, or the attributed scattering model of [30]. This technique may be used beyond SAR problems and into general engineering and scientific use.

5.1 Bayesian Methods

It is believed that improvements to shape detection can be made through the use of prior information, and this requires the use of Bayesian methods. In Section 3.2 we developed the bounds on the prior distributions based on the fact that the SAR phase history is sampled in both frequency and space. This leads to periodicity in the parameter space (and alias zones in SAR imaging) which must be avoided.

5.2 Numerical Integration

The detection of canonical shapes requires the use of a wide variety of mathematical techniques from statistics and numerical analysis. We reviewed techniques for computational integration of arbitrary functions in order to show their applicability to the SAR research community.

In Section 2.3 we explored the concept of the quadrature, whereby the integral of a function may be approximated by the sum of a set of weighted sample points (abscissas). We showed how the error in this approximation is related to how well the function can be approximated by a polynomial of order N. Further, the Gaussian quadrature defines the abscissas in terms of orthogonal polynomials, and this additional constraint leads to an increased accuracy of the integral approximation without the need for additional sample points.

For the purposes of this research, the Legendre polynomials were used to create the quadrature rules. These rules have finite bounds and the appropriate weight function for the integrals commonly encountered. The Gauss-Legendre quadrature was expanded from the simple 1-dimensional case to D-dimensional integrals with arbitrary bounds (by use of the affine transformation).

5.3 Probability Mass

The numerical integration techniques (particularly the Gauss-Legendre quadrature) can be used to calculate the denominator in Bayes' Rule. However, over the course of this research, we found a new use for this integral: the conversion of the posterior density $\boldsymbol{p}(\underline{\boldsymbol{\Theta}} \,|\, \underline{\boldsymbol{y}})$ into a discrete set of probability masses P.

We have shown that the probability mass is much more than just converting a continuous distribution into a discrete one. By defining the probability mass as the integral over a finite interval of the posterior density, areas of high density value (i.e. peaks in the posterior) are preserved regardless of the 'size' of the interval. Since the Bayesian detection and estimation process often looks for peaks in posterior space, this peak-preserving property is highly desirable.

We developed an algorithm whereby the posterior space can be partitioned into intervals (wide at first), and areas where a peak occurs are quickly found. These areas are then sub-partitioned into smaller intervals and the process is repeated. The result is a mult-zoom capability so that at any iteration only areas of ‘interesting’ probability information are examined. The results of Chapter 4 show how this mult-zoom capability can be used to quickly examine the fine structure of the posterior space.

The probability mass analysis at a given interval is also more than just a means of finding peaks. The results of Section 3.3.3 provide a formal statement and proof that the probability mass (the integral of posterior density over finite bounds) is a measure of the cost of an estimate. The probability mass transforms parameter estimation from a process of finding a best parameter estimate $\underline{\hat{\boldsymbol{\Theta}}}$ to a process of finding the nearest *range* of parameters. The cost function forces all parameter estimates within an interval to be equally correct. This new cost, and an associated new distance metric, creates a new type of maximum a posteriori (MAP) estimate that is tailored to use in a set of discrete probability masses.

5.4 Credible Regions

A technique for presenting a measure of confidence, the credible region, was adapted for the use with the probability mass. In Section 2.5, we presented the Gill definition of the credible region. This is a region where we can be assured that the ‘true’ parameter $\underline{\boldsymbol{\Theta}}$ has some probability α of being inside. The probability mass is a natural fit with credible region since both are statements about multidimensional integrals of the posterior density.

We presented two different approaches to defining the credible region: by fixed thresholding and by ranking the masses in order. Fixed thresholding is a natural extension of techniques based on the posterior density. In this approach, only those masses whose value exceeds the threshold γ are included in the credible region. The results of Section 4.6 show this threshold must change when the interval size changes, and the interval size is expected to change frequently in the multi-zoom parameter estimation.

An alternative approach, by ranking the probability masses, requires only the parameter α; it does not require a new threshold after each change in interval size. Instead, the masses (at that interval size) are ranked in order from largest to smallest, and the top M masses that add up to α are included in the credible set.

5.5 Implementation on GPU Hardware

For this research, we often encounter integrals of D parameter dimensions. The Gaussian quadrature of order N requires computation of the function at N^D points. For F frequency bins and K flight path samples a single quadrature requires FKN^D calculations of the Jackson phase history. This can be several billion calculations resulting in a single probability mass. The multi-zoom algorithm of Section 3.3.4 used in the creation of the results in Chapter 4 uses 20-by-20 grids of probability mass. Almost a trillion calculations must be performed to produce the plots in Chapter 4.

The Graphics Processor Unit (GPU) provides low-cost hardware to perform such calculations. In the GPU, there exist several hundred (or thousand) processors that execute the same code with different arguments. As a result, a single piece of C++ code is dispatched to all processors and each computes a portion (perhaps only one sample) of the phase history.

The CUDA framework is a driver for Windows and Linux that allows the capabilities of the GPU to be used by C++ programmers. Additionally MATLAB has support to call these code blocks (called CUDA Kernels) from within existing MATLAB codes. Over the course of this research, the Jackson phase history models were ported to CUDA codes. However the high-level sequencing and plotting remained as MATLAB code for ease of use.

In this research, algorithms that took over one hour in MATLAB using `for` loops and vector calculation were reduced to about 10 seconds. As a result the seemingly daunting calculations required in multi-zoom estimation and marginalization became practical to implement.

5.6 Future Work

This research has focused on new analytical, statistical and numerical techniques to estimate the parameters of shapes in a SAR target scene. We were successful at demonstrating the techniques in a controlled environment. However, there are many areas for future research.

5.6.1 Model Order Determination.

The multi-zoom estimation technique described in Section 3.3.4 requires knowledge of the total number of shapes in the scene. Without this information the algorithm will attempt to incorrectly model the difference between an estimated parameter instance and the observed signal as Gaussian noise. In either under-fit (too few estimated shapes) or over-fit (too many shapes) scenarios, the error is unlikely to be Gaussian. Particularly, it is unlikely to be independent, identically distributed (IID) and so all estimates of probability mass are in error. Therefore, an accurate model order is crucial. The results of [24] produce a technique for counting shapes based on the Jackson phase history that may be applicable.

5.6.2 Noise Sensitivity.

The posterior density is calculated assuming knowledge of the observed signal noise variance σ^2. Although this may not affect the MAP estimate to a great deal, changes in variance will scale the bounds of the credible region. Therefore, an accurate estimate of the variance is necessary for use of the credible region. Further research could focus on techniques for calculating the variance, and characterizing the sensitivity of the credible region to uncertainty in variance.

5.6.3 Developing Intuition about Posterior Space.

The graphs of probability mass shown in Chapter 4 are (we believe) the first attempt at truly exploring the posterior space (whether as a density or a series of masses) of the Jackson canonical shape models. This type of capability can allow further research into building models of the different coupling between parameters. Interesting areas for such exploration are the roll-pitch-yaw ambiguities, whereby combinations of roll, pitch, and yaw can produce the same phase history, and coupling between symmetric parameters such as plate height and length. Research into these areas can help build intuition about the limitations of the canonical shape framework and to provide ways of overcoming them. A real-time Graphical User Interface (GUI) that shows posterior space while the user adjusts parameters may be of particular value.

5.6.4 Improved Numerical Precision.

The results of this research were very encouraging and the plots of probability mass correspond to our intuitive understanding. In most cases, we expect to see a single peak in posterior space except where ambiguities such as the 2-plate case of Section 4.3 exist. However, in a large number of cases, the actual probability is below the numerical precision of the computing platform. This is particularly problematic when Gaussian probabilities are used, since the exponential for a long signal (many frequency and azimuth/elevation samples) can be very large. Both MATLAB and CUDA use the IEEE standard double-precision floating point number format, but in the course of this research there were numerous examples of needs to calculate factors such as $\exp(-10^8)$. Future research may focus on improving the technique to use the logarithm of the probability in order to preserve numeric results.

5.6.5 ATR Framework.

Finally, an extremely important area of future research is the development of a type of ATR framework that uses the posterior distribution directly (in the form of probability masses). The goal of this research is to develop techniques that detect the canonical shapes, so that these detected shapes may be re-assembled into complex objects like tanks, buildings, and structures. However most ATR techniques encountered over the course of this research focus on matching the received phase history to that of entire complex objects at once. Future research could focus on the use of the posterior masses to develop new prior distributions, and the problem of re-assembly of the canonical shapes into compound objects. The kernel-based learning methods of [31] could possibly be adapted to determine trends in the distribution of parameters that would lead to accurate prior distributions.

Bibliography

[1] J. A. Jackson, “Three-dimensional feature models for synthetic aperture radar and experiments in feature extraction,” Ph.D. dissertation, Ohio State University, 2009.

[2] G. B. Hammond II, “Target classification of canonical scatterers using classical estimation and dictionary based techniques,” Master’s thesis, Air Force Institute of Technology, 2012.

[3] D. Marsh, *Applied Geometry for Computer Graphics and CAD 2nd Ed.* Springer, 2005.

[4] S. Ghahramani, *Fundamentals of Probability, 2nd Ed.* Prentice Hall, 2000.

[5] S. M. Kay, *Fundamentals of Statistical Signal Processing*. Prentice Hall, 1993, vol. 1.

[6] H. Stark and J. W. Woods, *Probability and Random Processes with Applications to Signal Processing*. Prentice Hall, 2002.

[7] G. J. Feldman and R. D. Cousins, “A unified approach to the classical statistical analysis of small signals,” *Physical Review D*, vol. 57, no. 7, p. 3873, 1998.

[8] S. M. Kay, *Fundamentals of Statistical Signal Processing*. Prentice Hall, 1993, vol. 2.

[9] G. Evans, *Practical Numerical Integration*. John Wiley & Sons, 1993.

[10] A. R. Krommer and C. W. Ueberhuber, *Computational Integration*. SIAM, 1998, vol. 53.

[11] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*. Courier Dover Publications, 2007.

[12] V. S. Ryaben’kii and S. V. Tsynkov, *A Theoretical Introduction to Numerical Analysis*. Chapman & Hall/CRC, Boca Raton, FL, 2007.

[13] R. L. Burden and J. D. Faires, *Numerical Analysis, 7th Ed*. Brooks/Cole, Pacific Grove, CA, 2001.

[14] D. Zwillinger, *Handbook of Integration*. Jones and Bartlett Publishers, Boston, 1992.

[15] G. H. Golub and J. H. Welsch, “Calculation of Gauss quadrature rules,” *Mathematics of Computation*, vol. 23, no. 106, pp. 221–230, 1969.

[16] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions: With Formulars, Graphs, and Mathematical Tables*. Dover Publications, 1964, vol. 55.

[17] J. Gill, *Bayesian Methods: A Social and Behavioral Sciences Approach, 2nd Edition*. Taylor & Francis, 2008.

[18] S. Kay, *Intuitive Probability and Random Processes using MATLAB (R)*. Springer, 2006.

[19] W. Press *et al.*, *Numerical Recipes in C*. Cambridge University Press, 1992.

[20] R. A. Poisel, *Target Acquisition in Communications Electronic Warfare Systems*. Artech House, 2004.

[21] G. Hammond and J. Jackson, "SAR canonical feature extraction using molecule dictionaries," in *Radar Conference (RADAR), 2013 IEEE*, April 2013, pp. 1–6.

[22] N. Wilt, *The CUDA Handbook: A Comprehensive Guide to GPU Programming*. Addison-Wesley, 2013.

[23] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Computing*. Addison-Wesley, Upper Saddle River, NJ, 2011.

[24] M. Crosser, "Improved dictionary formation and search for synthetic aperture radar canonical feature extraction," Master's thesis, Air Force Institute of Technology, 2014.

[25] C. V. Jakowatz *et al.*, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach*. Kluwer Academic Publishers, Boston, 1996.

[26] B. W. Solverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[27] L. E. Jaulin, *Applied Interval Analysis: with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer, 2001.

[28] M. A. Ellis and B. Stroustrup, *The Annotated C++ Reference Manual*. Addison-Wesley, Reading, MA, 1990.

[29] A. Fog, "Calling conventions for different C++ compilers and operating systems," Technical University of Denmark, Tech. Rep., 2013.

[30] L. C. Potter and R. L. Moses, "Attributed scattering centers for SAR ATR," *IEEE Transactions on Image processing*, vol. 6, no. 1, pp. 79–91, 1997.

[31] M. Flynn, "Salient feature identification and analysis using kernel-based classification techniques for synthetic aperture radar automatic target recognition," Master's thesis, Air Force Institute of Technology, 2014.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)* 27–03–2014	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* Oct 2012–Mar 2014

4. TITLE AND SUBTITLE Bayesian Methods and Confidence Intervals for Automatic Target Recognition of SAR Canonical Shapes	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) RADEMACHER, RICHARD W., Captain, USAF	**5d. PROJECT NUMBER** LRIR12RY19COR
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-14-M-62

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr. Michael Kendra
Air Force Office of Scientific Research (AFOSR/RTA)
875 N. Randolph St. Suite 325
Arlington, VA 22203
michael.kendra@afosr.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

This research develops a new Bayesian technique for the detection of scattering primitives in synthetic aperture radar (SAR) phase history data received from a sensor platform. The primary goal of this research is the estimation of size, position, and orientation parameters defined by the "canonical" shape primitives of Jackson. Previous Bayesian methods for this problem have focused on the traditional *maximum a posteriori* (MAP) estimate based on the posterior density. A new concept, the probability mass interval, is developed. In this technique the posterior density is partitioned into intervals, which are then integrated to form a probability mass over that interval using the Gaussian quadrature numerical integration techniques. The posterior density is therefore discretized in such a way that the location of local peaks are preserved. A formal treatment is given to the effect of locally integrating the posterio density in the context of parameter estimation. It is shown that the operation of choosing the interval with the highest probability mass is equivalent to an optimum Bayesian estimator that places zero cost on a "range" of parameters. The range is user-controlled, and is akin to the idea of parameter resolution. Additionally the peak-preserving property allows the user to begin with coarse intervals and "zoom" in as they see fit. Associated with these estimates is a measure of quality called the credible interval (or credible set). The credible interval (set) is a region of parameter space where the "true" parameter is located with a user-defined probability. Narrow credible intervals are associated with high-quality estimates while wide credible intervals are associated with poor estimates. The techniques are implemented in state-of-the-art graphics processor unit (GPU) hardware, which allows the numerical integration to be performed in a reasonable time. A typical estimator requires several hundred million computations and the GPU implementation reduces the computation time from several hours to a few seconds. The mass interval estimation technique may be used on any Bayesian problem, but is demonstrated here using each of the canonical shape models of Jackson. The technique successfully estimates parameters in different scenarios including simple shapes, multiple shapes, incorrect shape (i.e. trying to estimate parameters using the wrong model). The results of this research are a new exploration of the posterior distributions of the canonical shape model, improved numerical integration strategies, and a new statistical technique for the Bayesian estimation of parameters.

15. SUBJECT TERMS

Radar, Synthetic Aperture Radar, Automatic Target Recognition, Bayesian Methods, Confidence Interval, Gaussian Quadrature, Numerical Methods, Parameter Estimation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr. Julie A. Jackson, AFIT/ENG
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)*
U	U	U	UU	116	(937) 255-3636 x4678 julie.jackson@afit.edu